UiO : Department of Informatics
University of Oslo

Design of a bilateral master-slave system with haptic feedback for ultrasound examinations

Jørgen Enger Fjellin

Master's Thesis Autumn 2013

Design of a bilateral master-slave system with haptic feedback for ultrasound examinations

Jørgen Enger Fjellin

1st August 2013

Department of Informatics
Faculty of Mathematics and Natural Sciences
University of Oslo
Oslo, Norway

The Intervention Center
Oslo University Hospital, Rikshospitalet
Faculty of Medicine
University of Oslo
Oslo, Norway

Abstract

A bilateral master-slave system with haptice feedback has been implemented and adapted to the use in ultrasound examinations. All hardware used is commercially available hardware that can be bought freely and used for a wide range of applications. User tests have been performed to assess how haptic feedback is contributing to the performance in such a system. Allthough some technical difficulties was experienced, the results are indicating that haptic feedback helps the operator to keep the slave more steady when in contact and to not apply unnecessary high force.

Acknowledgements

The Intervention Center at Oslo University Hosptial is a very exciting and motivating place to be. I would like to express my gratitude for having had the opportunity to work at this place, using the lab and equipment there. Thanks is in order to my supervisor Ole Jakob Elle who allways presented a different angle of perspective when I was stuck in the same chain of thoughts. Kim Mathiassen, my co-supervisor, needs a special acknowledgement and thanks for all the time he has spent helping me, teaching me to use the robot setup and to come up with practical solutions and discussing results. Without Kim, I do not know if this thesis would never have seen the light of day. Edvard Naerum deserves many thanks for taking time out of his busy schedule to come and discuss haptics with me. These have been fruitful discussions that has been much appreciated. An acknowledgement is also in order for Ngyen Ho Quoc Phuong for enlightening conversations about haptics in front of the coffee machine. I would also like thank the "gang" at computer-room 3 at the Intervention Center for many an interesting discussion over lunch and for making me feel welcome from day one. Finally, I would also like to thank my manager at SAS, Kim Andre Eidem, for showing me great flexibility at the office these last few months so that I would be able to finish this thesis.

Contents

List of Figures

List of Tables

Preface

People have been coming to the hospitals from far away in order to meet the expertise located at big central university hospitals. At the same time, the districts are having a hard time getting experienced and qualified personell to come and work in the rural areas. Also, local hospitals in the districts are merged or shut down, which means many people get even longer way to travel than they used to. Technology can be able to assist in these problems. This century, technology have allowed us to move an operating surgeon to another room, even the other side of the atlantic, whilst still operating on a patient.

Research on teleoperation is beeing done on many areas. This thesis is a part of an ongoing research at The Intervention Center, Oslo University Hospital on a semi-autonomous robotic system for use in medical diagnostics and treatment. This is the PhD-study of Kim Mathiassen, which this thesis will be a sub part of. In this study, a Universal Robot UR5 is equipped with an ultrasound probe and the aim is to use this for diagnostics and treatment.

The focus of this thesis will be the implementation of a master-slave system with haptic feedback. This system will in turn be adapted and implemented into the extend framework developed by Kim Mathiassen around the UR5 robot. The system will then be test on volunteers to see what effects the adding of haptic feedback has on such a system. The research will be based on earlier research of PhD Edvard Naerum at The Intervention Center.

Part I

Introduction

Chapter 1

Introduction

At the intervention Center at Oslo University Hospital, Rikshospitalet research is taking place on automation within the field of medical robotics. One of the ongoing projects is a semi-automation of ultrasound diagnostics. Radiologists performing an ultra-sound examination tends to get musculoskeletal disorders induced by their work [51]. During an examination the radiologist needs to apply the ultrasound probe with a certain static force to the patient for a good image. Often this force is applied in awkward positions that can be harmful to the users arm. Over time, this can trigger disorders like carpatel tunnel syndrome [63] and general wear that renders pain to the operator. This can eventually make him/her unable to do these kind of jobs. By setting up a semi-autonomic teleoperator to do the actual job you will be able to free the operator of the situations that are inducing tendinitis[1], and wear. Adding a 6 degrees of freedom controlling device with haptic feedback, it would be possible for the operator to feel what he is doing without applying that much force; the force can be scaled. Another possibility when introducing such a system is to make the position and orientation relative, which makes it possible to reach awkward angles without having the operator work in the exact same angle.

1.1 Previous work

Teleoperation has been around for a while, and much research is taking place on this matter. For the last one and a half decade teleoperators designed for extracorporal ultrasound diagnostics and treatment have been reaserched and developed [45]. Gourdon *et alumni* developed a custom robot specifically for ultrasound diagnostics in 1999 called SYRTECH [19]. This robot is hold in place over a patient while an operator does the teleoperation. Succesfull clinical tests with this setup has been performed [6]. In 1999 Salcudean *et alumni* also developed a robot for ultrasound diagnostics [51]. This however was designed more like a robotic arm that was mounted at the patient side and was then able to

[1]Tendinitis - Inflammation of tendons.

reach the patient within it's workspace. [25] made an ultrasound robot where the patient would sit or stand upright whilst being examined. Courreges *et alumni* presented the OTELO system in [15] which can be seen as a third generation of the SYRTECH system. As it's predecessor, this robot too needs to be placed at the patient before the teleoperation can start. [31] is another parallell designed ultrasound robot which is designed to be put on the patient. In [64] Vilchis *et alumni* proposed a robot that is spanned over the patient's bed at four places. These bands are then all connected to a wrist-like construction in the middle where the ultrasound head is. All is controlled by a Phantom device from Sensable Technologies. The TERMI robot is yet another custom built robot setup for ultrasound diagnostics [65]. Pierrot *et alumni* proposed the hippocrate robotic system with ultrasound in [42]. Here they uses an existing industrial robot arm, however no haptic feedback is implemented.

As can be seen there are many varieties of robotic systems for ultrasound diagnostics undergoing research and development. Not many of them have haptic feedback, and those who do seems to have a self-made custom setup; either a custom slave, master or both. By developing things from scratch it becomes possible to design the exact specifications that is desired. However it is costly and time-consuming to develop something from scratch. Using commercially available equipment can be benefitial as these products are mass produced and has the corresponding price. Another benefit of having a setup with more general hardware is the possibility to use it for different means than was originally planned. By changing the tool on a robot arm it becomes much more useful than a custom design tailored for ultrasound diagnostics.

In this thesis robotic and haptic devices that are commercially available will be used to set up an ultrasound examination system. Thus it will be possible to assess whether custom designed teleoperators are necessary for ultrasound robotic systems or if commercially available hardware can be a viable option.

1.2 Research motivation

The ultimate goal with the research in the field of haptics within the medical community must be to come up with a bilateral teleoperated system that is completely transparent and that can reproduce the stiffness of the environment on the slave side of the system in the master end of the system. There could be different varieties for different purposes (neurosurgery, minimally invasive surgery, ultrasound diagnostics, perhaps even gastroscopy or endoscopy with the possibility of palpation and more).

There are several challenges along the way there. Developing a force-torque sensor that can be commercially produced and that meets the criteria posed earlier are one challenge. Pursuing the force-estimating algorithms to see how they cope with robots with higher dimensions of freedom is another way to go. Then there is the force-sensing part. Setting

up a setup with a robot equipped with a force-torque sensor, then implementing it with a haptic device and see how it works in regard to transparancy and see what can be done in order to optimize the system. The framework will be the same for different master-slave systems in regard of the utilization of the sensor data. Transferring that framework to another setup will simply mean changing the right parameters in the algorithms in order to adapt it for the next system.

Setting up a system with haptic feedback means having to invest in a suitable haptic device and develop the necessary software to implement it in the existing system. Haptic devices are sophisticated technology that costs money. Before investing in expensive equipment the necessity of the investment must be established. Will haptic feedback contribute in a positive way to a bilateral master-slave system designed for ultrasound examinations, or is the money better spent elsewhere? This thesis will implement such a system, perform user tests to evaluate the user's experience and compare this to the same system without haptic feedback. This way, a difference between having and not having haptic feedback should be discovered and the necessity of further research and investement in equipment for this matter assessed.

Chapter 2

Background

The invention of the word 'robot', is a Czech playwriter named Karel Capek who in 1921 wrote a play called "Rossum's Universal Robots"[68]. The word 'robot' is the stem of the Slavic word *robota*, which means 'self labour'[68]. The next time robots showed up was in the writings of the famous science fiction writer Isaac Asimov, where they appeared in the 40's[13]. Then in 1956, George Charles Devol Jr. was granted the first patent for a digitally programmable robot arm[9]. This would later turn into the Unimate, which was the worlds first industrial robot, who started to work at General Motors' diecasting plant in Trenton New Jersey[9]. 5 years later, General Motors opened it's state-of-the-art plant in Lordstown Ohio with nearly 100 Unimates at work, making it possible to produce 110 cars an hour which was twice the rate as any other car factory[9].

2.1 Teleoperation

The definition of the word *teleoperator* is: "Any remote-controlled machine which mimics or responds to the actions of a human controller at a distance" [43]. Remotely operating a machine to do a specific task has a very wide area of application. Much of the motivation for remote controlled robots to do tasks instead of doing them oneself comes from the wish of operating in dangerous environments. This might be a burning house, a radioactive area, not only a nuclear plant but also in hospitals where x-rays are widely used. Think about the operators that stands beside a patient undergoing fluoroscopy, CT or patients that simply need to take x-rays during surgery. These operators, allthough protected by ledclothing, will inevitably be exposed to an unhealthy amount of x-rays over time. Other hazardous areas might be a collapsed house, a mine field, the bottom of the sea, space, a volcano or other very remote places[35].

There is an extensive use today allready to use remotely controlled robots. The United States' army have UAV's (Unmanned Aerial Vehicle)that are remotely controlled by pilots sitting safely within the USA while the uav's are flying on the other side of the world, doing reconnaissance, bombing targets or other needs their mission mith require [66].

Figure 2.1: *Dextre robot on the International Space Station. [40]*

Another very hostile environment is space. Astronauts takes a huge risk every time they need to go for a space walk in order to fix something on the exterior of a space vessel. By applying a teleoperated master-slave robot into space, it will be possible to do this both more safely and more effective[48]. One can also imagine using teleoperated robots to build and assemble space stations or bases on the moon or other planets as it is very unpractical to send a large building crew to such hazardous construction sites[23]. If you then prebuild large modules on earth and send up, you can imagine having a large construction teleoperator in space that someone will use while assembling the space station. By scaling the movements up and down to fit different teleoperators, the human operator can sit safely on earth and doing probably all the job from the same control. Already there is a teleoperated robot in orbit around our planet at the International Space Station (figure 2.1). The canadian robot Dextre is attached to the exterior of the space station, and can be used for inspection or replacement of orbital replacement units [8]. An orbital replacement unit is any kind of hardware that is readily replaceable. This robot is already today beeing remotely controlled from earth.

The last two decades we have seen robotics moving from laboratories into operating theatres worldwide. These new robotic systems give the surgeons the means to perform more complex procedures. These days a number of procedures are performed using a minimally invasive approach[29]. The da Vinci© surgical system is the only commercially available one as of today. One of the reasons why these methods are gaining increasing popularity is because of the effect it has on the patient.

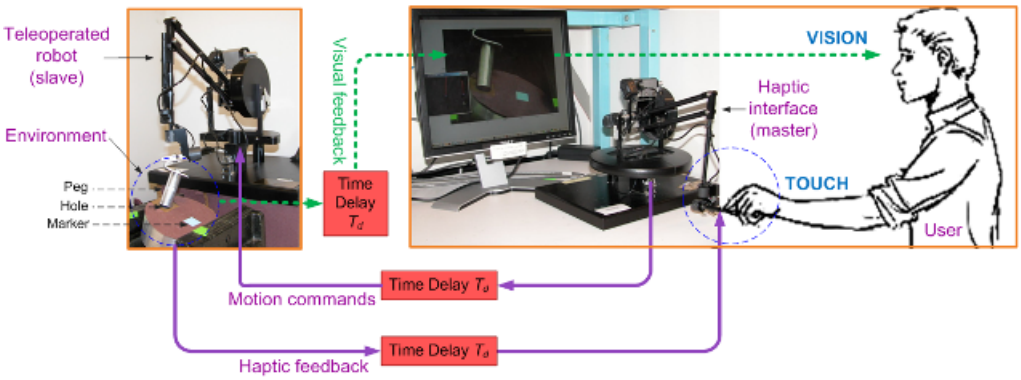


Figure 2.2: *Figure of a teleoperation master-slave system with haptic feedback. [59]*

Performing surgery this way can reduce procedure time, especially when suturing. As an effect of using the minimal invasive technique hospital stay is usually shortened. A shortened hospital stay means that the patient is recovering quicker, which is good for the patient but also means less cost of treating the patient. In other words, hospitals will be able to treat more patients at the same amount of time.

2.1.1 Teleoperation with delay

As the distance between the master and the slave increases, inevitably the communications delay will increase. This is a major challenge to both the stability and the usability of a bilateral teleoperating system. There has been many experiments with both surgery-like tasks and actual surgery on in vivo animals. The first experiment involving a real patient was a transatlantic teleoperation was in 2003, they performed telesurgery on a woman in Strasbourg, while the surgeon was situated in New York [10]. This experiment did not take Haptic Feedback into consideration, but mainly focused on whether or not it would be possible to do telesurgery over such a distance. They found that as long as the total communication delay was not more than 330 ms, they managed to cope with the delay. All above 330 though, turned out from difficult to impossible. This paper, [10] came out in 2003. Since then, many more have performed telesurgery over great distances in clinical trials with lab-animals [5] [47], and even clinical surgery [3].

2.2 Haptic feedback

The word "haptic" comes from greek άπτικ-ός (able to come in contact with) and is described in the Oxford English Dictionary as meaning: *Of, pertaining to, or relating to the sense of touch or tactile senseations* [44]. One of the human's senses is the sense of touch. We interact with the world by feeling the structure of things, feeling the strength of things,

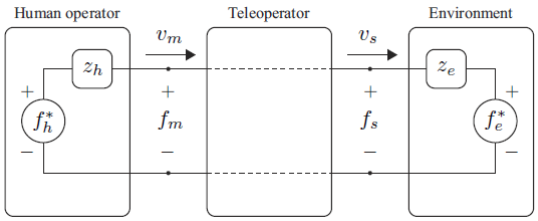


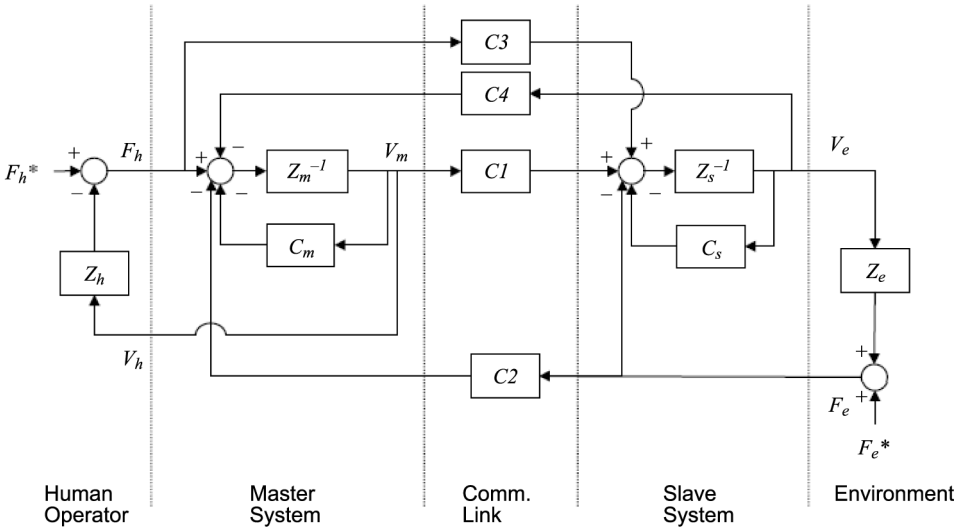
Figure 2.3: *Figure of a two-port network model [38].*

manipulating things by applying a force to the world around us. These days we interact with haptic interfaces several places. Our smartphones vibrate when we touch them so that we know that our touch is registered, our controller rumble and vibrate when we are playing video games.

When it comes to teleoperation for medical use, haptic feedback has yet to be implemented in commercially available products. In medical use, accuracy and precision is of extreme importance. In order to implement a good haptic feedback, you need to know how much force is applied from the apparatus you are controlling onto the environment in which it is operating. Within the area of medical use, there are many different environments that the robotic manipulator will be operating in. These environments ranges from palpating tissue tissue or being in contact with the skin for an ultrasound diagnostic session to operating within the abdominal or even catheterization during angioplasty[1] [32]. Usually the teleoperator is set up as a master-slave system[4] where the human operator is controlling the master part of the system and then the masters movements are copied by the slave part and it starts manipulating the environment. In such a system, the ideal case would be when the forces felt by the human operator are the same as the forces applied on the slave-environment. In such a case the human operator will feel like he is manipulating the environment directly, and he will not feel the electronics inbetween at all; the system is transparent.

Such a master-slave system can be modelled as a two-port network model [38]. One can model a teleoperator system like this because it manipulates and interacts with the environment through two "ports": The human operator in the master-end of the system and the environment in which it is supposed to operate in the slave-end. In figure 2.3, we see the line-out of such a system. The impedance of the human operator and the environment is denoted by the z_h and z_e, and ditto exogenous forces denoted by f_h^* and f_e^*. As we can see on this model, this system is transparent when $f_m = f_s$ and $v_m = v_s$ at all times and for all frequencies [38]. A linear two-port system can be completely characterized by equation 2.1, where H is a 2×2 hybrid matrix that relates the different variables (the h_{ij} in the hybrid matrix are rational functions of the laplace variable s)[38]:

[1]Angioplasty - Widening narrowed or obstructed blood vessels mechanically from within.



Source: Lawrence (1993)

Figure 2.4: *Figure of a four-channel Lawrence Architecture.*

$$\begin{bmatrix} f_m \\ -v_s \end{bmatrix} = H \begin{bmatrix} v_m \\ f_s \end{bmatrix} \qquad H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \tag{2.1}$$

From this equation we can see that a system is transparent when the hybrid matrix is [38]:

$$H = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{2.2}$$

This means that we need to have a way of exchanging information between the master and the slave system. There are many ways we could do this. If we do not think about the haptic feedback for a minute, we could imagine a simple one-way communcation from the master to the slave. This would be called a *one-channel* communication. The master would then send the position in which the master was positioned and the slave would have some control mechanism that immediately sought to get in the right position. This would of course not take into account the amount of force applied, and would simply give all it had got in order to get to the desired position.

In *two-channel* communication we get two channels between the master and the slave. Either we could use these to send two different types of information one way, or we could use these channels to send information in both directions. In order to get haptic feedback, we need information sent in both directions. Then there is the matter of what information to pass on through these channels. We can send information about position, speed or force. Edvard Naerum *et al.* has shown in their research that transparency is in fact possible to achieve with only two channels [38]. This can only be done by sending complimentary signals, meaning we have to send force in one direction and velocity in the other.

Relating this to the Lawrence Architecture (figure 2.4) this means having either gains $c1 = c2 = 0$ or $c3 = c4 = 0$ in order to achieve transparency with two channels. Having $c1 = c2 = 0$ is called a forward-efforct controller, while having $c3 = c4 = 0$ is called a forward-flow controller.

For *three-channel* and *four-channel* it is evident that transparency can be achieved, as complimentary signals will allways be exchanged between the master and the slave. [38]

2.2.1 Force measurement

A key element to haptic feedback is knowing how much force is applied onto the environment by the slave manipulator. There are in general two ways we can aquire that information. Either we can estimate the forces applied by means of measuring other values, like angle and current in joints, or we can place a force sensor at a suitable location of our master/slave.

Force can be described as a vector in 3 dimensional space. Any n-dimensional vector can be decomposed to 3 vectors lying on the axis of it's worlds coordinate frame. There is also rotational force, called torque, which is defined as the cross product between the force applied to an area of an object and the distance to it's mass center; $\tau = rxF$. In order to measure these forces, we need a force-torque sensor. According to [20], commercial force sensors can be largely divided into three types based on the measurement mechanism: strain-gauge type, piezo-electric type, and resistive type. The strain-gauge type for sensor is popular due to it's high sensitivity. This type of force sensor consists of an elastic element attached to a strain-gauge. Then from the deformation of the elastic element, the force causing the deformation is calculated. Since the strain-gauge is attached to the elastic element, it will too be exposed to the forces at hand. Thus a mechanical limit is necessery to implement to protetct the strain-gauge in case the sensor is overloaded, which makes these sensors more expensive since they are more difficult to design. Finally the signal levels that comes out of this are so small that they need an amplifier to be able to read the outcome. The piezo-electric type force sensor are somewhat similar when it comes to the sensitivity and the amplifier. Use of an amplifier makes both these systems noisy and bulky. The resistive type of force sensors, also known as force sensing resistors, are a polymer thick film device which exhibits a decrease in resistance with an increase in the force applied to the active surface.

Forces are not only present upon contact. If you have a setup with a robotic arm that is moving with high velocity through free space, and there is a force/torque sensor attached to it's tip, that sensor will experience forces while the arm is moving through the free space due to inertia. It will experience forces like the coriolis force, the sentripetal force and the force of gravity from different angles. A study [26] points at the fact that these forces might be stronger than the actual contact-force so that the operator won't notice if he/she comes in contact with something on the move. They propose a way to remove or scale down non-contact forces from the haptic

information given to the operator by implementing a 6 degrees of freedom accelerometer to calculate these forces and then subtract them.
Force-torque sensors for such a setup is widely accessible. One thing one must bear in mind with sensor measurements is that sensors in general are a bit noisy and have a limited bandwidth[39].

For a system intended for medical use there are more challenges. If they are to be used on a system like the da Vinci, the sensors needs to be small enough to fit down the trocars of a minimally invasive surgery setup. They also need to be able to work in the environment within a patient and they need to be easy to clean/sterilize[46]. As of today, there are no commercially available force torque sensors that meets these criteria, but a few prototypes as well as some concepts exists [53] [46] .

Since it is not necessarily trivial to implement a force-torque sensor into a robotic surgical system, estimating of these forces is a possible option[39].

2.2.2 Force estimation

There are many ways to estimate the forces applied to the environment by a manipulator, for example [61] and [41]. The forces applied can be split up in two parts: the force that acts normal to the tissue surface, and the momentum generated by a force applied tangential to the tissue surface; the torque. To start of we need to derive the dynamics of our robot system. Deriving the dynamics of a robot-system is often done by the use of Lagrange's equation [36]. The equation will have the following form for an n degrees of freedom robot [36]:

$$M(q)\ddot{q} + C(q,\dot{q})\ N(q) = \tau \tag{2.3}$$

Here, M is the inertial matrix of the system, C is the coriolis matrix and N is the gravity-vector [36]. These can all be computed analytically, but some of these values are unknown in the sence that they change during the operation of the robot. This is for example link lengths, masses and moment of inertias [36]. Thus we can seperate these out into an parameter of their own, which we will call θ, and our new equation will then look like this [36]:

$$Y(q,\dot{q},\ddot{q})\theta = \tau \tag{2.4}$$

Y is here a $n \times p$ regressor matrix and θ a vector of p dimensions, where p is the number of unknown parameters [36]. This is the ideal case. Of course our world is not ideal, and all the joints does have some internal friction as well. This friction is dependent of the speed of the joint, thus it is denoted by $\tau_f(\dot{q})$ [36]. So this is our final equation:

$$Y(q,\dot{q},\ddot{q})\theta + \tau_f(\dot{q}) = \tau \tag{2.5}$$

Knowing this $\tau_f(\dot{q})$ is the tricky part. Friction as a concept is not fully understood, and most of our knowledge of friction constants are found

empirically. This can be done by implementing amongst other things neural networks [56] and wavelet networks [37].

One of the benefits of estimating the forces is it that you need to make a model of the system. This means that model can be reused in a simulation, wchich in turn can be used in a surgical simulator so that surgeons and medical students becoming surgeons can practice surgeries outside of the operating room[14] [67].

Chapter 3

Commercially available systems

3.1 The da Vinci© Surgical System

The da Vinci© is, as allready stated, the only commercially available robotic surgical system for the time being. According to Intuitive Surgical[58] the da Vinci© consists of mainly three components; the surgeon's console, the patient-side cart and the 3-D vision system. In 2010 about 270 000 procedures were performed by the da Vinci© system alone[58]. The da Vinci© was also installed in 1752[58] operation theatres worldwide as of 31st of december 2010. The da Vinci© is also FDA-approved for a number of procedures; general laporascopic[1], non-cardiac thoracoscopic[2] , prostatectomy[3], cardiotomy[4], cardiac revascularization[5], uroligic surgical, pediatric surgical and transoral otolaryngologic[6] surgical procedures [58].

3.1.1 Main components

The surgeon's console is where the surgeons are seated. The newest edition, the da Vinci Si© actually comes with two surgeon's consoles. These consoles are equipped with two handles and a 3-D viewer. The handles are the controls the surgeons use to steer the instruments that are inside the patient. The handles, as well as the instruments, have 7 degrees of freedom [57].

The patient-side cart is the cart where all the electromechanical arms are mounted. This cart is movable, and it is positioned beside the operating table according to the procedure that is about to take place. The cart has a total of three arms (four in the newest Si version). One arm is

[1]Laporascopy - Operation in the abdominal regian through small incisions.
[2]Thoracoscopy - Procedures in the thoracic cave, behind the chest bones.
[3]Prostatectomy - Removal of the prostate.
[4]Cardiotomy - Medical procedure where an insition is made to the heart
[5]Cardiac revascularization - Restoration of blood vessels and flow to the heart muscle.
[6]Otolaryngocology - The medical specialization in Ear, nose and throat.

reserved the endoscope, while the rest of the arms are attached with the different instruments available.

3-D vision system is composed of a special endoscope with dual camera for HD 3-D vision.

EndoWrist is the name of the design behind the instruments you can attach on the da Vinci© arms. It is called EndoWrist as it is designed to have 7 degrees of freedom. The degree of freedom and design is chosen as it is comparable to our own hands, which in turn gives the surgeons the same amount of freedom as if they were in there themselves, only with the abilitiy to move in a much smaller scale [24]. A variety of different instruments exists that you might need during a surgery, and these are all possible to change during an operation [58].

3.1.2 Advantages and disadvantages

As with everything, there is no such thing as a free lunch. Yes, there are many advantages when it comes taking such a system like the da Vinci© into use, but there are also some concerns. Murphy *et alumni* has come up with a list of advantages and disadvantages regarding the da Vinci© compared to non-robotic laparoscopic surgery [34]. Let's look at the advantages first.

3-D visualization. With the da Vinci© the surgeon gets a 3D view of the inside of the patient, as well as an output on HD screens for the spectators and other personell in the room.

The "fulcrum effect". When operating laparoscopically you encounter the "fulcrum effect". In laymen terms, when a surgeon operates with a miminally invasive method, his movements is inverted in the inside of the patient because of the fulcrum effect from the patients abdominal wall [17]. As is seen in figure 3.1. A robotic surgical system like the da Vinci© can cancel this effect from the surgeons point of view. Although the significance of this is still unknown, the learning time for a surgeon without any laparoscopic experience may be shorter for a minimally invisive surgery using the da Vinci© as opposed to one using the conventional laparoscopic way.

Motion scaling and elemination of tremor. By applying a telemanipulator it is possible to scale down the motions of the surgeon, which makes it possible for the surgeon to perform more precise surgery. If you apply a filter on top of that you will get rid of the tremors that comes using your hands in awkward positions for prolonged time, and the result is smooth and precise movements within the patient.

Reduced fatigue. As opposed to conventional laparoscopic surgery where the surgeon stands beside the patient for the entire operation, which may last for hours, he gets to sit down by the console while operating. This in turn reduces the risk of operating "error" during a long and complicated operation.

Now let's look at the concerns. First of all it is rather expensive. $1.44 million was the average price of a da Vinci© system in 2010 [58]. It also

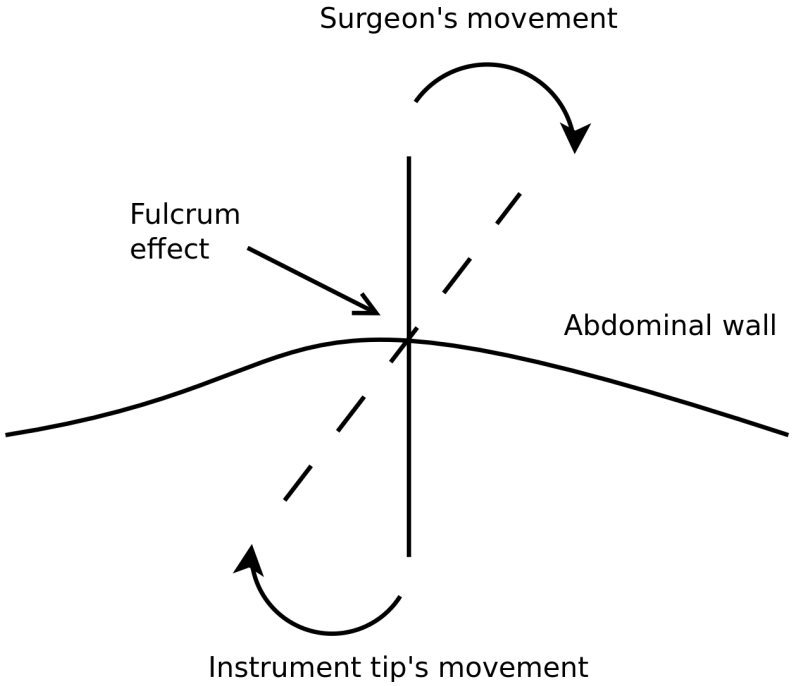


Figure 3.1: *The fulcrum effect.*

requires some annual service, which was on average $143,000 per system [58].

There is at the moment no haptic feedback in this system, which means that the surgeon looses his sense of touch when taking this equipment into use. This sense is of great importance to a surgeon as applying to high a force might create unnecesarry damage to the patient. It is also an important source of information to him. In open surgery a surgeon can detect wether a tissue is healthy or inflamed simply by palpating it. When operating with a robot today, an experienced surgeon judges the state of the tissue by observing the deformation of it. [62]

3.2 Haptic devices

As of today, several companies offer commercially available haptic devices. They come in different shapes and sizes which suits different needs. I will now go through a couple of different haptic devices, sorted by the producing company. All the information about the different devices is found from the producers website. An overview of the specifications for the different devices can be found in appendix A.

3.2.1 Force Dimension

Force dimension[16] is a swiss company that produces 3 series of haptic devices, with a variety within each series of degrees of freedom. They have the sigma series, the omega series and the delta series. The sigma series

Figure 3.2: *The Force Dimension's Sigma*

Figure 3.3: *The Force Dimension's Omega 6*

comes with only one product, the sigma 7. The sigma 7 can be seen in figure 3.2. This device has 7 degrees of freedom with both force and torque feedback. This device is aimed for the aerospace and medical industry, according to their product page. The omega series comes in three varieties. The omega 3, 6 and 7. The number in these reflects the degrees of freedom in each device. The omega 3 has 3 active translations, the omega 6 has 3 active translations as well as 3 active rotations and finally the omega 7 has a grasping extension in addition to the translation and rotation. The omega 6 can be seen in figure 3.3. Then the delta series consists of two models, the delta 3 and delta 6 which also has 3 translations and 3 translations + rotations. The delta series is somewhat larger than the omega series, and thus have a larger workingspace. A delta 6 can be seen in figure 3.4. Another interesting feature of the devices of Force Dimension is the fact that they have used a parallel manipulator design for their haptic devices.

3.2.2 Haption

Haption[22] is a french company that delivers 4 series of products. These are the Virtuose (fig. 3.5), the MAT (fig. 3.6), the Inca (fig 3.7) and the Able. Able is a haptic interface more than a device. This interface is

Figure 3.4: *The Force Dimension's Delta 6*

Figure 3.5: *The Haption's Virtuose*

designed for using on your arms, and thus is this product formed like an exoskeleton. You wear it, and get the workspace of your entire arm. The Able comes in three configurations with 4, 5 and 7 degrees of freedom. The Inca is a cable driven device with 6 degrees of freedom, and it needs a fairly large space. It is based upon the SPIDAR of professor Makoto Sato at the Precision and Intelligence Laboratory at Tokyo Institute of Technology [52]. The MAT also fairly large. This haptic device has 6 degrees of freedom and is a large arm coming down from above you. It has a workspace of about a quarter of a circle down from it's mount-point. The last series from this producer, the Virtuose is the only desktop-model from this producer. It comes in three products, one with only 3 degrees of freedom, and then two with 6 degrees of freedom. These two are different sized.

3.2.3 Geomagic (former Sensable technologies)

Geomagic is an american company within the 3D business. They are focusing on software for working with 3D modeling and simulation, as well as hardware for 3D printing and manipulation. In april 2012 [18] they aquired what was then known as Sensable Technologies, a company

Figure 3.6: *The Haption's MAT*

Figure 3.7: *The Haption's Inca*

specializing in haptic devices and 3D interaction software aimed CAM/CAD and simulation of physical interaction for it's haptic devices. As this is a rather recent event, most of the literature denotes these products buy their old names. I will therefore stick to the old naming naming when speaking of products originating from Sensable technologies. Sensable[54] was an american company situated in Wilmington. They had a line of four haptic devices, all within their Phantom series. There is the Phantom Omni, Phantom Desktop, Phantom Premium and Phantom Premium 6DOF. All of these haptic devices are desktop sized. The Phantomn Omni comes with 6 degrees of freedom. It is not very big, thus it has a very compact workspace of 160 x 120 x 70 mm. The Phantom Desktop is slightly smaller than the Phantom Omni, but it has a slightly bigger workspace of 160 x 120 x 120. As with the Omni, the Desktop has 6 degrees of freedom. They both have a pen-like end-effector which makes it easy to use for 3D-painting or modelling. The Phantom Premium comes in four varieties; Phantom Premium 1.0, 1.5, 1.5HF (High Force) and 3.0. The difference between these four types is mainly size and strength, where 3.0 is the largest and strongest while 1.0 is the smallest and weakest. They all come with 3 degrees of freedom and force feedback as standard, but it is possible to get an encoder stylus so that you get tracking in 6 degrees, meaning you get both position and rotation. The Phantom Premium 6DOF is basically the Premium 1.5 and 3.0 with full force feedback in all 6 degrees of freedom. It is also possible to get a 7th degree of freedom by adding a tool snap-on on the end effector, for example scissor handles. This 7th degree do not have haptic feedback. The table below depicts the old versus the new product names. The haptic devices can be seen in figure 3.8, 3.9, 3.10 and 3.11.

Old name	New Name
Phantom Omni	Geomagic Touch
Phantom Desktop	Geomagic Touch X
Phantom Premium X.X	Phantom Premium X.X

Table 3.1: *Geomagic and Sensable name conversion.*

3.2.4 Butterfly Haptics

Butterfly Haptics[21] is a rather interesting concept for a haptic device. As opposed to "normal" haptic devices relying on a mechanical link, Butterfly Haptics has chosen to use magnetic levitation. This device can get either 6 degrees of freedom. The handle that the user is holding is attached to a "flotor" that is floating in the magnetic field. The position and orientation of this "flotor" is tracked by optical sensors. A 7th degree of freedom, like for example a grip, is possible to add to the handle. The butterfly haptics can be seen in figure 3.12.

Figure 3.8: *The Sensable's Phantom Omni*

Figure 3.9: *The Sensable's Phantom Desktop*

Figure 3.10: *The Sensable's Phantom Premium 3.0*

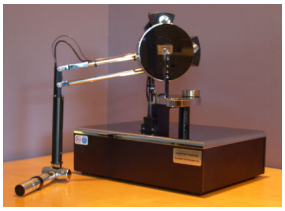


Figure 3.11: *The Sensable's Phantom Premium 1.5 6DOF*

Figure 3.12: *The Butterfly Haptics' device*

Part II

The Project

Chapter 4

Hardware Setup

In this project I am working with a Universal Robot UR6 and haptic device Phantom Omni from Sensable technologies.

4.1 Universal robot UR5

The UR5 is a 6 degree of freedom robotic arm from a danish producer named Universal Robots. The robot consists of 6 rotatable joints which can all rotate +/- 360 degrees. The robot has a weight of 18.4 kg, and can operate with a payload of 5 kg, a force of 49 N. The workspace of the UR5 is a part of a sphere with a radius of 850 mm, depending on how it is mounted. The robot is delivered with a computer that has low-level access to the robot. This controls the power set on the joint-actuators and performs necessary kinematic caluclations. The robot comes with a touch screen and a graphical user interface where the robot can be programmed and steered (figure 4.1). An ethernet interface is available for programming the robot from an external environment.

4.2 Controller computer

The computer is running an Intel i7-960 processor, which is a quad core processor running on 3.2 Ghz. It is equipped with 6GB of 1600 Mhz RAM as well as an SSD-drive and a SATA disk-drive. The computer is set up with Linux, a Debian distro version 6.0.6. The distro is running on kernel 2.6.38.8 compiled with Xenomai, a real-time framework integrating support for real-time computing within the Linux environment, guaranteeing deterministic computing.

4.3 Haptic device

There is a variety of different haptic devices that one can get hold of when choosing a haptic device. Takeing teleoperation of the ultrasound equipped robot arm as a use case, some devices will be better suited than others. In order to use an apparatus around patients, it needs to be reliable

Figure 4.1: *Universal robot UR5, with control interface [49]*

and safe; accurate. A sufficient workspace is also required for our haptic device. A haptic device like the one from Butterfly Haptics would probably be a bad idea for this use case due to the little workspace in lateral directions. As the robot does have 6 degrees of freedom, it would be preferable that the device controlling it also has 6 degrees of freedom. Preferably with force feedback for all 6 degrees. Based on these criteria, the Delta 6 from Force dimension would be recommended for the setup with the universal robot at the Intervention Center. The Delta 6 has 6 degrees of freedom, with force feedback in all of them. By having a parallel design, the actuators working to apply the force feedback are distributed between several actuators at once, as opposed to for the haptic devices with serial design. This way the maximum force feedback is more evenly distributed throughout the workspace, and do not have a large locationbased variance. The Delta 6 also have a reasonably large workspace. A comparison based on data sheets have been done in order to recommend this particular haptic device. The specifics for this device can be seen in table 4.1. The comparison table can be found in appendix A.

Although the Delta 6 from Force Dimension has been recommended, the Phantom Omni has been chosen for this setup. This was chosen due to the fact that this was the haptic device available at the Intervention Center robot laboratory. The Phantom Omni is a small haptic device from Sensable Technologies. It has a workspace of 160x120x70 mm and a nominal resolution of 0.055 mm. It has a maximum exertable force of 3.3 N when it's arms is in an orthogonal position. Force feedback is available in 3 degrees of freedom, and position sensing is availbale in 6 degrees of freedom. The cartesian position, e.g. the 3 first degrees of freedom, is

Producer	Force Dimension
Product name	Delta 6
Degrees of freedom	6
Force Feedback	6 DOF
Maximum volume	400x260 mm
Translation workspace	Sphere
Maximum force	20 N
Maximum torque	150 mNm
Minimum resolution translation	0.01 mm
Miminum resolution rotation	0.04 degrees
Interface	USB 2.0

Table 4.1: *Force Dimension Delta 6 specifications*

Figure 4.2: *Sensable Phantom Omni, image from Dentsable website. [60]*

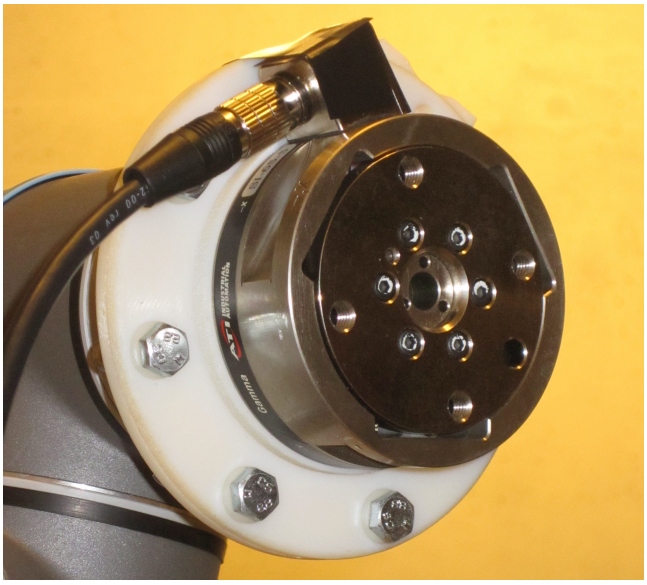


Figure 4.3: Force-torque sensor attached to UR5 robot.

measured with digital encoders. The stylus gimbal, giving roll, pitch and yaw, is measured through rotational potentiometers with linear encoding (Figure 4.2). During an ultrasound session gel is applied at the surface that shall be investigated. This gel makes surface friction very low, so we can say that the contribution from surface to the torque is virtually none. Thus a device with force feedback in only three degrees is found to be sufficient.

4.4 Force-torque sensor

The Universal robot UR5 is equipped with a force-torque sensor. This is a Gamma DAQ Force-Torque transducer SI-65-5 from ATI Industrial Automation, 1031 Goodworth Drive, Apex NC 27539 - USA . The Gamma force-torque sensor is connected to the computer via a DAQ - Data Aquisition card in a PCI - slot in the computer. A picture of the force sensor can bee seen in figure 4.3, and it's specifications in table 4.2 [7].

Sensing range				Resolution			
F_x, F_y	F_z	$\tau_x, \tau y$	τ_z	F_x, F_y	F_z	$\tau_x, \tau y$	τ_z
65 N	200 N	5 Nm	6 Nm	1/80 N	1/40 N	10/13333 Nm	10/13333 Nm

Table 4.2: Force sensor specifics

4.5 Connection

The force/torque sensor is connected to the controller computer through a high speed data aqusition card. The Phantom Omni is connected to the controller computer through a firewire interface. The Universal Robot

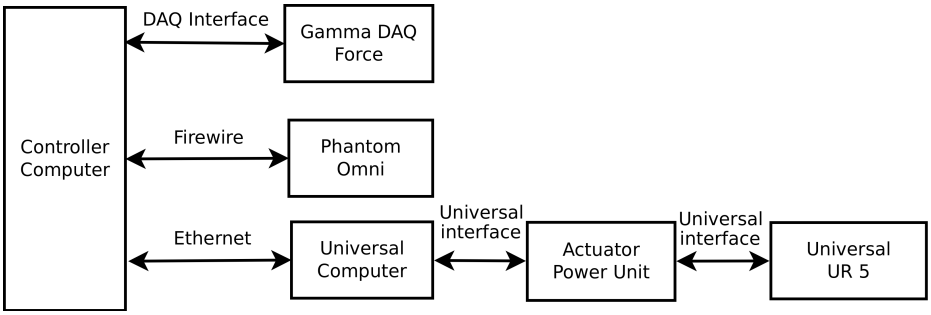


Figure 4.4: *Layout of connected devices and computers*

computer is connected to the controller computer through an ethernet interface. This is in turn connected to a power unit through an interface from Universal Robots, which in turn sends the power to the robot. Joint angles is passed on through the same communcation channel. The setup layout can be seen in figure 4.4

Chapter 5

Implementation

All software is implemented in either C or C++. C is a low-level programming language readily available on all platforms. C++ is considered an intermediate language; C with the possibility of object oriented programming. A linear algebra library for C++ is also installed and used. Labview is a graphical programming language from National Instruments making it easy to prototype graphical user interfaces and visualizing data in real-time. It is also supporting data aquisition from National Instruments' data aquisition hardware.

5.1 Existing framework

A block chart of software layout can be seen in figure 5.1. All the columns is running in seperate threads. The haptic controller thread is the one implemented in this work.

5.1.1 In-house framework extension

The inhouse extension of the framework is developed at the Intervention Center. A new set of more intuitive and general calls are made in a wrapper that goes around and extends the api that comes from the robot producer. This extension makes it possible to program and command the robot in a real-time environment, which is critical if it is going to be used in a clinical setup. A real time environment is a deterministic environment, which as an Operating System means that a process are guaranteed a predetermined time on the central processing unit. This new wrapper is then programmed into a daemon that runs several real time threads which performs different tasks from sending and receiving messages to the robot, to the reading of the force-torque sensor. The robot is updated in a 125 Hz frequenzy. External programs can send commands to the robot by sending commands to the daemon through a real time-queue. This can then be commanded in several ways; by position, by velocity or by force. A front-end in Labview is also made which let you easily command and configure the robot on an ad-hoc basis. Here you also

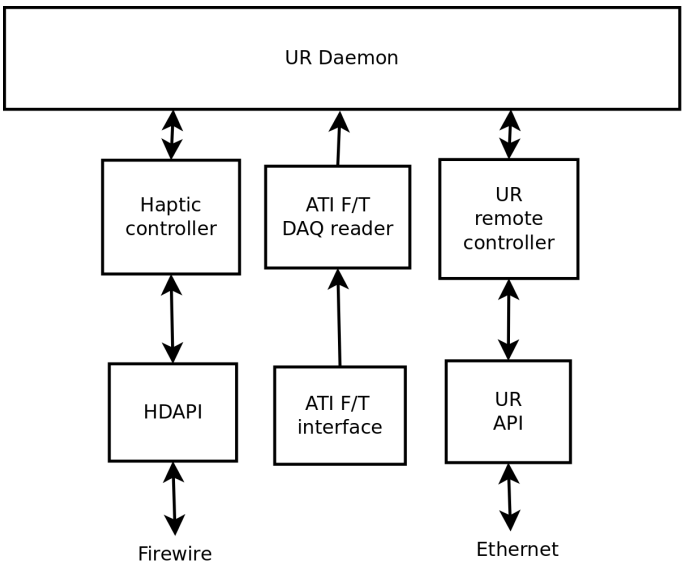


Figure 5.1: *Layout of implementation of software framework.*

select which controller is to be used. By implementing this in-house extension to the framework a real-time system was achieved.

The force torque sensor will inevitably have a bias due to gravity and it's own mass as well as anything attached to the toolframe of the slave. This bias needs to be calculated in order to get useful readings from it. In this case a method by [12] is used that involves sampling the force at different positionts and angular velocities, preferably around a unit axis, and on the basis of these samples calculating the center of mass and the mass of the object at hand. The force sensor is also suffering from noise. An elliptic noisereducing filter is thus implemented. The elliptic filter is chosen due to it's very good properties regarding phaseshift and frequency response.

5.1.2 Omni device communication and control

Sensable has a library called OpenHaptics which is available with their products. This is a library with calls to the haptic device that lets you read and write values from the device's registers. This library is not just for controlling the haptic device, but also for simulating and visualizing an environment. The library are split in two parts, which are named respectively HDAPI and HLAPI. HDAPI is written in C and is the interface for low-level communication and control of the haptic device. HLAPI is also written in C, and is the part for simulating and visualizing 3D environment, with haptic feedback. This is wrapped around HDAPI in order to simplify programming. In this work, only HDAPI has been used.

5.2 System modeling

In this design, there has not been made any dynamic model of neither the master nor the slave. This is done on purpose as part of the research. Due to small masses and velocities, it is assumed that forces due to inertia and coriolis effect can be neglected. Thus a simplified controller is designed. By making this assumption and implementing this kind of design, the outcome is a control architecture that is independent of what devices are plugged in at both ends. In figure 5.2 a simplified sofware layout of the implementation of the haptic controller is visualized. In the code, only a few methods is involved in the actual communication with the master and slave. The output of these are cartesian coordinates and forces. Thus it is possible to change the master or slave simply by changing the content of a few methods to the new API and recompile the code. The key to this independent controller design is to operate in cartesian coordinates and forces.

5.2.1 Two port network model

As earlier stated in the introduction, a master-slave teleoperating system can be modelled as a two-port network [38]. In [38] it is also stated that it is possible to obtain transparency in a teleoperating system with two communication channels by implementing the forward flow Extended Lawrence Architecture as seen in [38] (figure 5.3). This is a control architecture that sends velocity from the master to the slave and force from the slave to the master. This specific architecture was chosen for two reasons. First of all teleoperation with haptic feedback based on force measurement from the slave was desired. This ment that we needed a controller like the forward flow Extended Lawrence Architecture. which sent force from the slave to the master. In addition, this was already proven by [38] to be transparent. Another benefit with this controller is that it operates with a minimum of needed bandwith in the communications layer, only two channels. If this setup is to be implemented over a network some time in the future, this is a clear advantage. With these criterion, the forward flow control architecture stood out as a good choice.

5.2.2 Filtering

When the robot is supposedly standing still, it is never quite at ease. It is adjusting it's joints and coping with gravity which means there can be some small movement in the joints at all time. These small movements will manifest themselves as forces in the force sensor, which means there will be some vibration noise from the system at all time. The same thing is true when it comes to the human operator. It is hard for a human to hold a static position over a long period of time, thus there will be som small vibration noise on the haptic device as well. By implementing a low-pass filter it is possible to filter out this noise, as well as any unwanted spikes

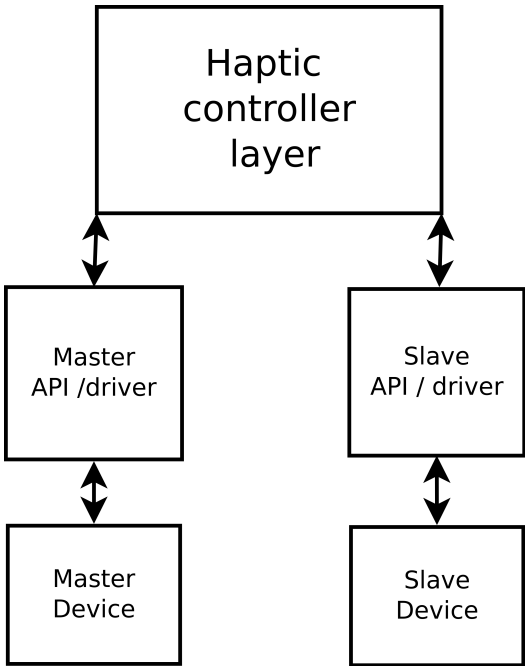


Figure 5.2: *Plot of layering of the proposed controller.*

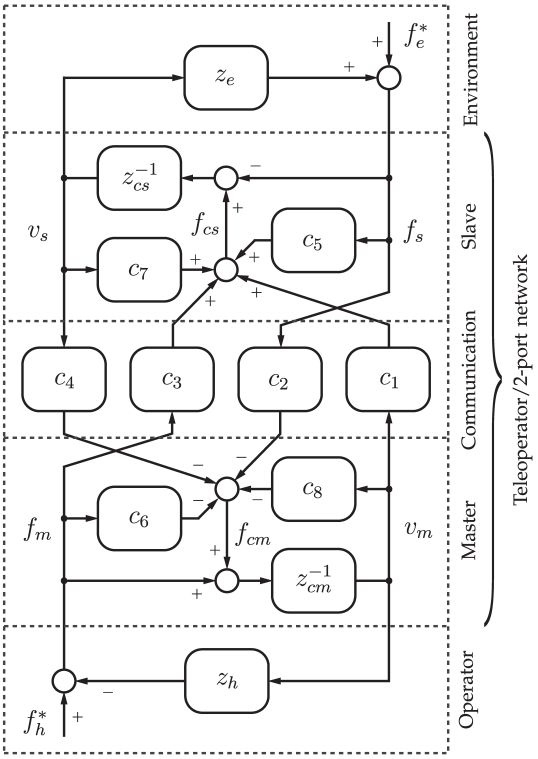


Figure 5.3: *The Extended Lawrence Architecture. In the forward flow setup gains c3 and c4 are set to zero.*

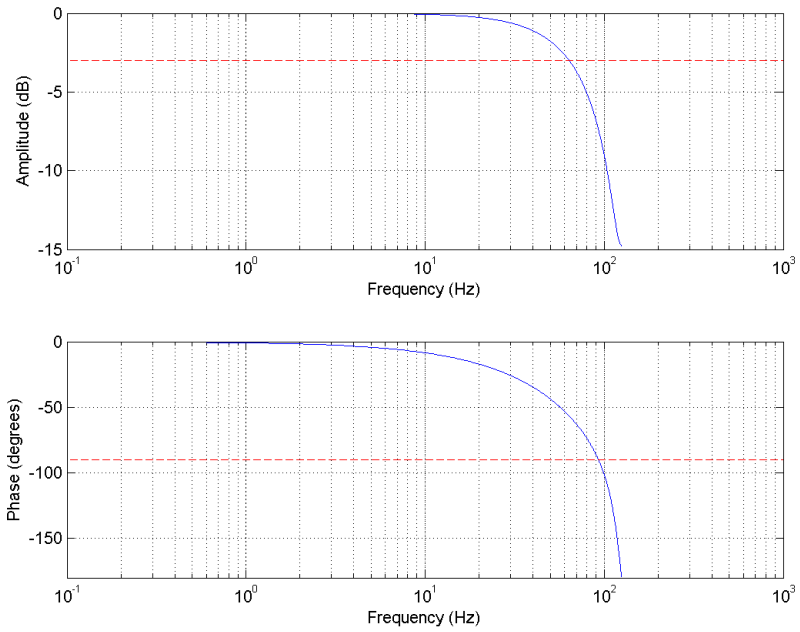


Figure 5.4: *Properties of implemented low-pass filter*

from the force sensor. In general, a system teleoperated by a human operator will mostly consist of low frequencies. However high frequent spikes from the force measurement might occur at the time of contact.These are unwanted, and will be effectively removed by a low pass filter. The low pass filter has been implemented with a cutoff frequency of 55 Hz and a periode of 3 samples. The filter response can be seen in figure 5.4. This was applied to both the velocity readings from the master and the force readings from the slave. However, for the force readings, this was not enough to cancel out the necessary noise. Thus an additional moving average filter of 3 samples were added, lowered the noise adequately. By haveing to moving averages right after one another the signals are in effect dampened. readings after the low pass filter. The properties of the moving average filter can be seen in figure 5.5, and the combined effect in figure 5.6. As can be seen in the figure 5.7 the dampening in much steeper than in the other plots, so we can expect a dampening in the force output.

5.2.3 Coordinate system transformation

In this setup the workspace of the master and the slave is different in size, the slave workspace is much larger than the master workspace. In order to use the entire workspace of the slave, mapping of the master workspace to the slave workspace would be needed. However it does not make any sense to mirror the workspaces completely. By making the position of the slave relative, we can send velocity commands to the slave and reuse the workspace of the master many times. The phantom omni has a button

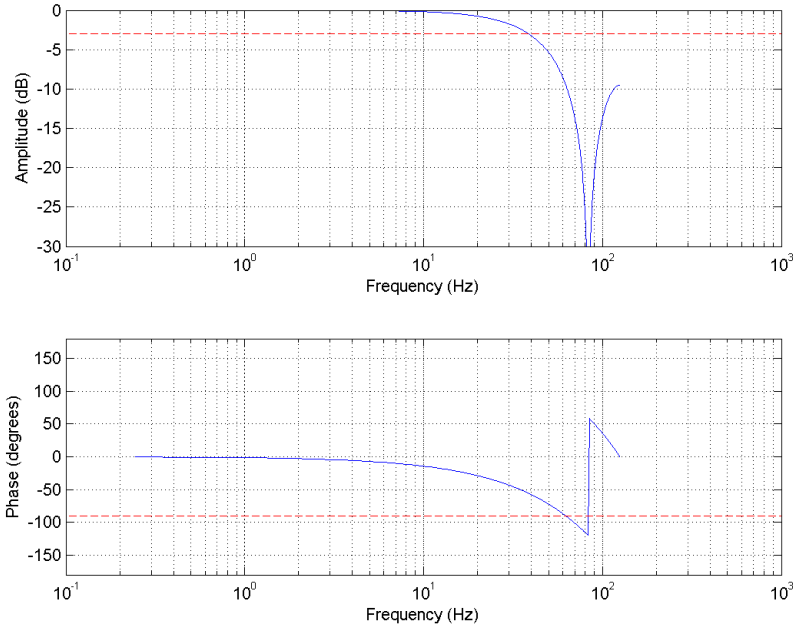


Figure 5.5: *Properties of implemented moving average filter*

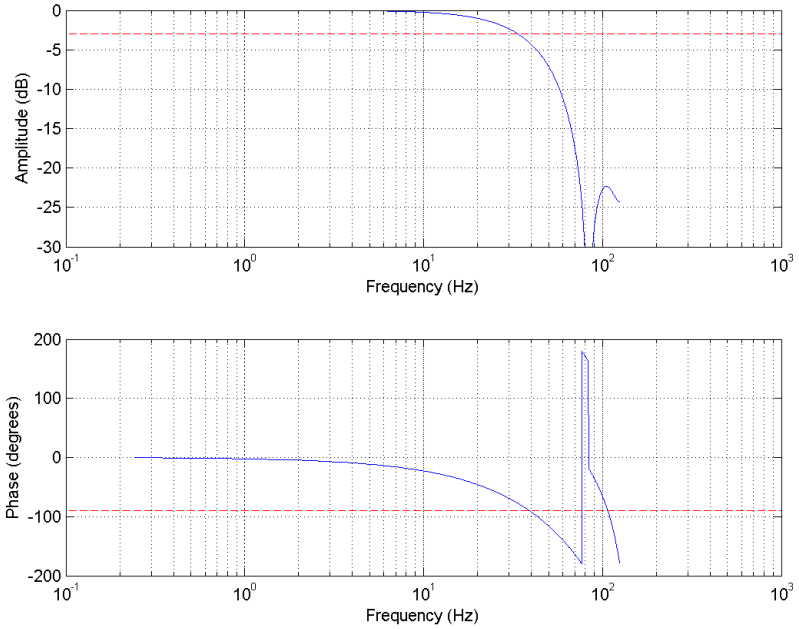


Figure 5.6: *Properties of combination of filters from figure 5.4 and 5.5.*

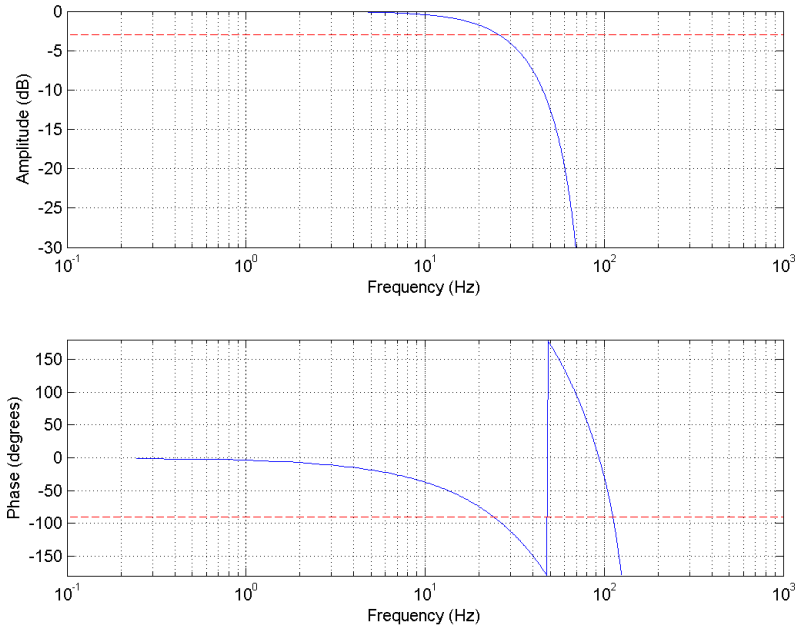


Figure 5.7: *Properties of combination of filters for force readings from figure 5.6 and 5.5.*

readily available for the index finger, and by simply applying this as an on/off switch for the teleoperation we are able to reuse the workspace. This is true for both the cartesian and angular velocity.

Ultimately velocity readings from the phantom omni was not possible to get, even though API from the vendor was used. However, current and last position was possible to get as well as the orientation matrix. Thus it was possible to derive the speed of the phantom by simply deriving the position. In order to get the relativeness regarding the orientation however, some more calculation was needed.

When teleoperation is not initiated, e.g. when the button is not pressed, the current rotation matrix of both the master and the slave is sampled and stored in a struct. When the button then is pushed, the rotation matrix will no longer be stored. Thus, we have starting rotation matrix R_0 of both our master and slave. We call this the master and slave bias, denoted R_{Mb} and R_{Sb}. When we move the master around we will have a relationship between the master and the master bias that we denote R_Δ. This is the relative orientation which we want to transfer to the slave. The relationship can be shown like this:

$$\begin{aligned} R_M^M R_\Delta &= R_{Mb}^M \\ (R_M^M)^{-1} R_M^M R_\Delta &= (R_M^M)^{-1} R_{Mb}^M \\ R_\Delta &= (R_M^M)^{-1} R_{Mb}^M \end{aligned} \tag{5.1}$$

The calculated orientation of the slave will then be:

$$R_S^S = R_\Delta R_{Sb}^S = (R_M^M)^{-1} R_{Mb}^M R_{Sb}^S \tag{5.2}$$

In this particular setup, the base coordinate system of the phantom omni and the base coordinate system of the UR5 are different, so it is necessary to apply a transformation around R_Δ. When choosing the common coordinate system, several options was available. It was chosen to use the slave's base coordinate system as the common system as this is where the interaction with the environment takes place. Another option would have been to use the toolframe as the common coordinate system. Effects and consequences of this choice will be further discussed in the discussion section. For the robot base coordinate frame, the following transformation matrix had to be used:

$$R_M^S = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{5.3}$$

The orientation of the slave will eventually calculated by the following equation.

$$R_S^S = (R_M^S)^{-1} (R_M^M)^{-1} R_{Mb}^M R_M^S R_{Sb}^S \tag{5.4}$$

Then we have the position as a vector p and the velocity as a vector v, both 3-dimensional vectors. For the sake of clarity we can denote them p_s and v_m in order to clearly signal which belongs to the master and slave. By the help of these, we make the two augmented matrices P_S and V_M as:

$$P_S = \begin{bmatrix} I_3 & p_S \\ 0 & 1 \end{bmatrix} \qquad V_M = \begin{bmatrix} I_3 & v_M \\ 0 & 1 \end{bmatrix} \tag{5.5}$$

Again we need to transform the velocity matrix to the slave coordinate system. Thus we transform V_M with the augmented transformation matrix T_R.

$$T_M^S = \begin{bmatrix} R_M^S & 0 \\ 0 & 1 \end{bmatrix} \qquad P_{set} = P_S T_M^S V_M (T_M^S)^{-1} \tag{5.6}$$

This leads us to the desired transformation matrix we want our slave to have regarding translation P_{set}. Making another augmented matrix from our already calculated rotation matrix R_S^S, we can get the Transformation matrix T_{set} which we use to command the slave.

$$T_S^S = \begin{bmatrix} R_S^S & 0 \\ 0 & 1 \end{bmatrix} \qquad T_{set} = P_{set} T_S^S \tag{5.7}$$

The joint velocities necessary to get to the desired setpoint, T_{set}, is then calculated by the kinematic library for the robot, and a p-controller calculates necessary joint speeds.

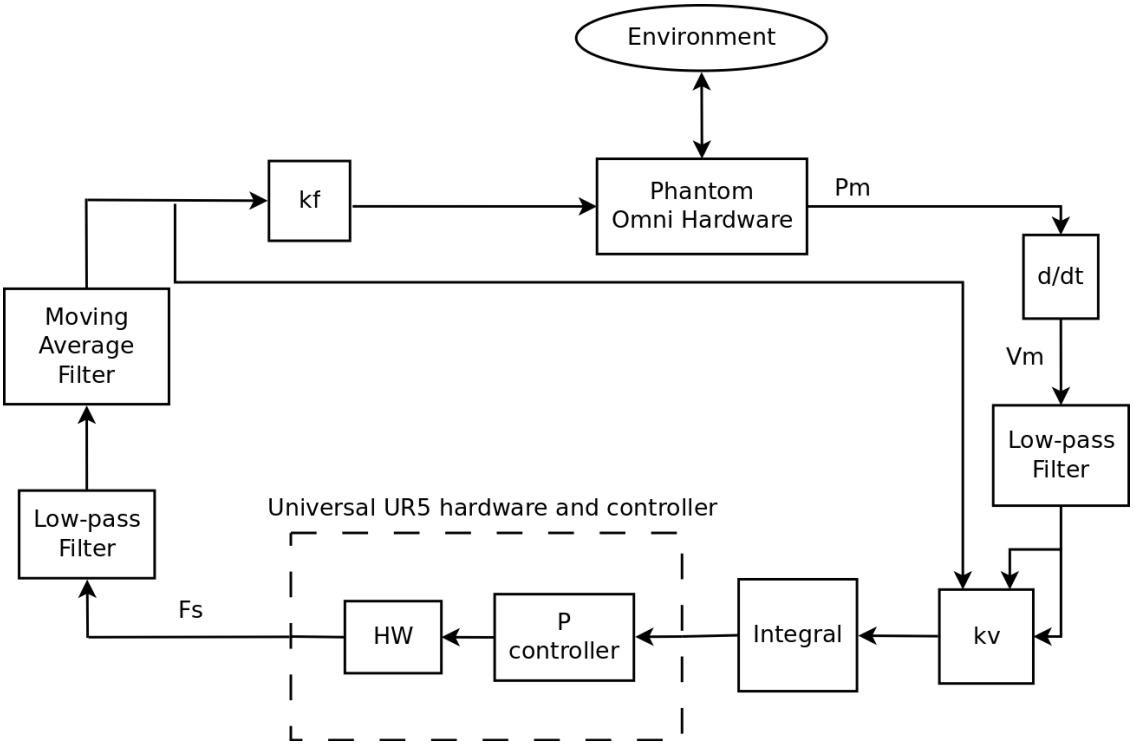


Figure 5.8: *System diagram of proposed haptic controller.*

5.2.4 Gain computation

As previously stated, one of the reasons motivating this research is to spare the ones performing ultrasound diagnostics from the wear of the work. This means that we do not necessarily want *full* transparancy, as we would the be faced with the same problem as before. In other words, we would like to scale the force sendt back to the operator in some manner. As stated by [38] a completely transparent teleoperation system does not guarantee stability. This means that we have to make a compromise between transparency and stability best suited for our needs.

In the literature there is usually one set of controllers doing the job, usually with dynamic gains in order to stabilize the system. These are designed to work in all situations. Some does not interfere until some conditions are met [50], while others work all the time in different scales. Since we have allready established that we in this case are not necessarily after true transparency a proposal is made to take changeable gains into use. It is easily imagineable that one set of gains is working better for teleoperation in free space than another that might work better for teleoperation in contact with the environment. Assuming that the force sensor, when calibrated, gives an output of 0 when not in touch with the environment, it follows that the slave is in contact with the environment when $|f| > 0$. Of course in the real world with noise, vibration and inertia it is improbable that the force sensor output is going to be 0 when moving around in free space. Thus to be sure that we are in contact we will need to

give the following criteria:

$$|f| > |N_{MAX}| \tag{5.8}$$

Where N is the noise in the system. This noise can either be remaining noise after filtering, or the raw input from sensor, as needed for the current system at hand. In this implementation, the filtered values are chosen as N_{MAX} was very large for the raw force input. By identifying the contact with the environment, it is possible for us to change our controller parameters, e.g. our gain, in order to have gains better fitted to the situation. Four different controllers are implemented in order to be tested. At the same time the gains k_v and k_f are introduced, which corresponds to c1 and c2 from the Extended Lawrence Architecture. The system is implemented like the block diagram in figure 5.8, where k_v and k_c are the gains getting changed in the different controllers. Due to the difference in size and strength between the master and the slave, as well as the workspace mapping, global constant gains for force and velocity are implemented. These are allways applied and comes in addition to the ones introduced beneath. In the ideal case, this corresponds to the following H matrix [38]:

$$\begin{bmatrix} f_m \\ -v_s \end{bmatrix} = \begin{bmatrix} 0 & k_f \\ -k_v & 0 \end{bmatrix} \begin{bmatrix} v_m \\ f_s \end{bmatrix} \tag{5.9}$$

Transparent controller

In the transparent controller, we want full transparency. This means that the force felt in the master should be the same as in the slave, and the velocity in the slave should be the same as for the master. To achieve this we need both our gains to be 1. Thus we get the following gains:

$$\begin{aligned} k_v &= 1 \\ k_f &= 1 \end{aligned} \tag{5.10}$$

Static Low controller

The static low controller is applied upon environment contact. It contains a constant k_c and is applied upon touch. In my setup, this is only applied to k_v, so that:

$$k_v = \begin{cases} k_c & \text{If } |f| > |N_{MAX}| \\ 1 & \text{If } |f| \leq |N_{MAX}| \end{cases} \qquad k_f = 1 \tag{5.11}$$

By lowering the speed we will get more accurate movements while we are in contact with the environment, e.g. the patient. It is imaginable that this k_c is an adjustable parameter that the operator can adjust regarding to his/her needs during operation. $|k_c|$ should be less than 1 in order to get sufficient damping of the system. The reason why this is only applied to k_v is that we want to keep the feeling of contact as accurate as possible.

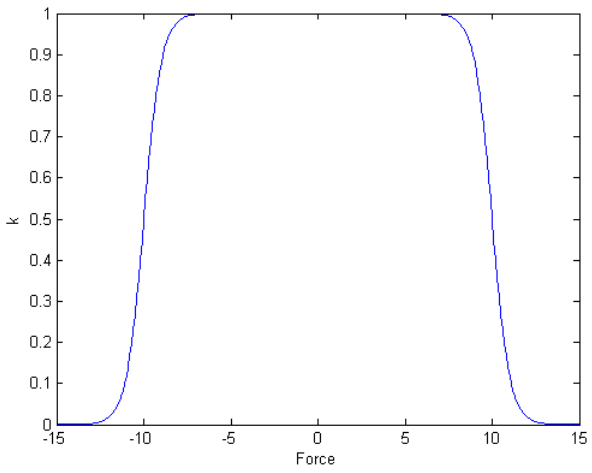


Figure 5.9: *Graph of the properties of the sigmoid function with* $\beta = 10$.

Sigmoid zero

If this setup is to be used in a patient examination, a built in limit as to how much force can be applied on the environment would have to be implemented. A study [51] has shown that the forces applied during an ultrasound diagnostics session lies between 4-7 N. Thus it is unnessecary to allow forces stronger than that to be applied on the environment. A dynamic gain for the velocity to the slave that is correlated with the force exertion onto the environment is proposed. The gain k is calculated as follows:

$$k = \frac{1}{1 + e^{2(|f|-\beta)}} \tag{5.12}$$

Which is a gain with properties as can be seen in figure 5.9. β is a bandwidth variable, which controls the force range before velocity is lowered. The formula has been developed by visualization in order to get the desired gain properties as seen in the figure. This gain ensures that as the forces starts to get adequately strong, the velocity commanded to the slave will converge to zero and so the exertion of unnessecary high forces onto the environment is avoided. This gain however makes it possible to get "stuck" within the high forces since the velocity converges to zero. In order to avoid this we need to turn the gain on an off depending on which direction we are going. If we are going towards the environment or not. This is easily computed. When we move towards and into an object, the force and velocity on the same axis will be oppositely directed. This means that $fv < 0$ if we are going towards the object, and $fv > 0$ if we are moving from it. Thus we can construct our gain like this:

$$k = \begin{cases} \frac{1}{1+e^{2(|f|-\beta)}} & \text{if } fv < 0 \\ 1 & \text{if } fv \geq 0 \end{cases}$$

Which gives us the following gains:

$$\begin{aligned} k_v &= k \\ k_f &= 1 \end{aligned} \tag{5.13}$$

Sigmoid low

The sigmoid low is almost the same controller as the sigmoid zero controller. The difference is that instead of tuning the velocity all the way down, it is dynamically going down to a predefined minimum. In a way this is a combination between the sigmoid zero and the static low controller. The computation follows:

$$k = \begin{cases} \frac{1-k_c}{1+e^{2(|f|-\beta)}} + k_c & \text{if } fv < 0 \\ 1 & \text{if } fv \geq 0 \end{cases} \tag{5.14}$$

$$\begin{aligned} k_v &= k \\ k_f &= 1 \end{aligned} \tag{5.15}$$

5.2.5 Modified β for experiments

Since the experiments is taking place on a phantom with rather low stiffness, it is probable that the force will not be large enough in order for the proposed sigmoid controllers to have any effect during the experiment. It is therefore implemented with a slight alteration in β, which makes the bandwdth narrower in order for the gain to take effect sooner. In the experiments, $\beta = 4$.

Chapter 6

Experiments

In order to test the system, a phantom is made to simulate organic tissue (figure 6.2). The phantom is made from gelatine, corn-flour and water. 3 table spoons of corn-flour and 9 plates of gelatine is sufficient for 400 ml of water in order to get the desired stiffness. This is in turn poured into a pai-form, where a few items have been placed so that the phantom will contain something that will be visible on ultrasound (figure 6.1). K is estimated by the equation for spring constant:

$$F = k\Delta p, \qquad k = \frac{F}{\Delta p} \tag{6.1}$$

The robot is equipped with a 3D-printed model of of an ultra-sound probe in it's toolframe (figure 6.6). By programming the robot to get the model in contact with the phantom and then program it to go deeper into into the phantom we can sample info on position and force. Using these samples we can calculate k. The different plots can be seen in figure 6.3, 6.4 and 6.5. As the phantom lies in the XY plane of the robot base system, it is only necessary to sample the position and force on the z-axis. In figure 6.3, the scale is adjusted so that the z-value 0 equals the surface of the phantom. As we can see, there are large variations when it comes to k. When Δp is very small, k becomes very large in our equation. We also see that the k drops at the end of our sampleing. This corresponds to the fact that at the end of this recording the surface broke at the position of the slave, and thus the force naturally went down. In order to find our k it is wise to calculate an average over a subset where k is not at it's extremeties. At sample 4550 we have the following values:

$$z = -0.001222 \qquad f = -0.915 \qquad k = 748.6 \tag{6.2}$$

The f is that large already due to force bias and noise. During this estimation, the surface of the phantom ruptured. By this plot it is easy to estimate where it happened, at about sample 10580. Taking the average of k between these two found samples we find our spring constant: $k = 533.4$.

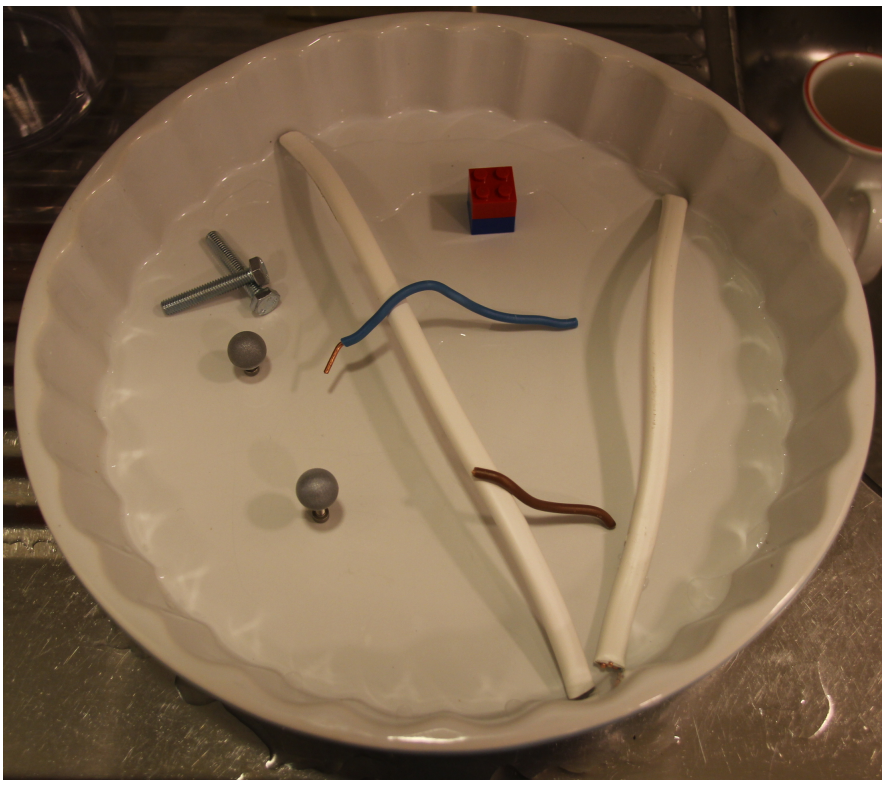


Figure 6.1: *Contents of phantom*

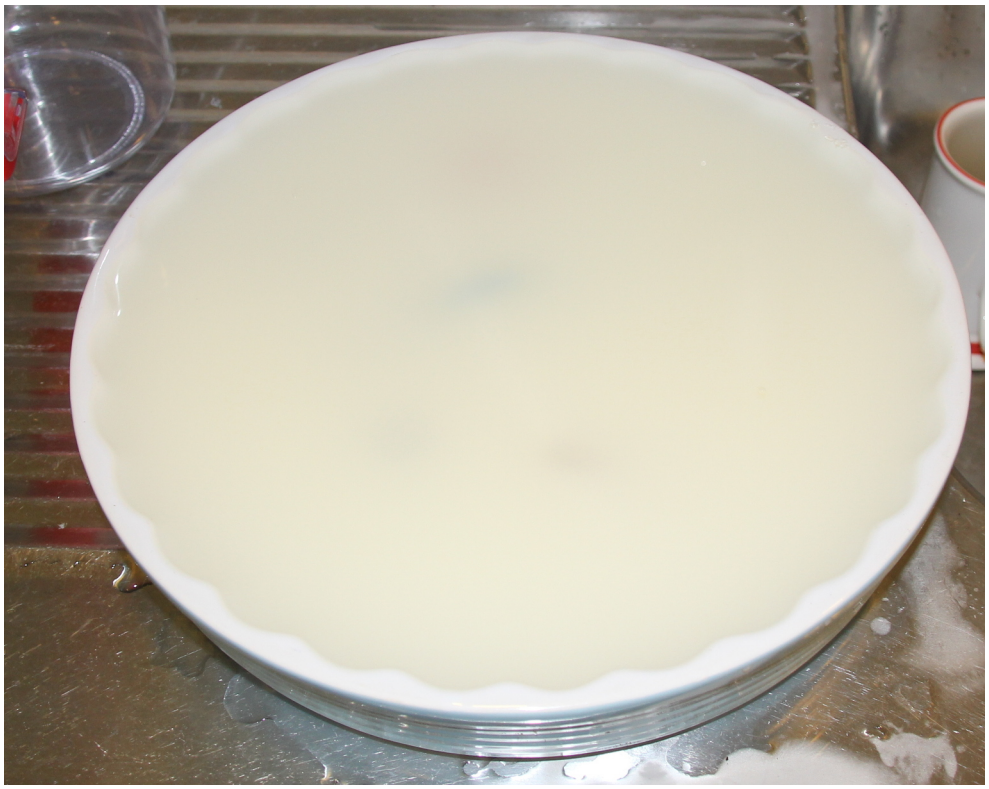


Figure 6.2: *Phantom done, but still warm.*

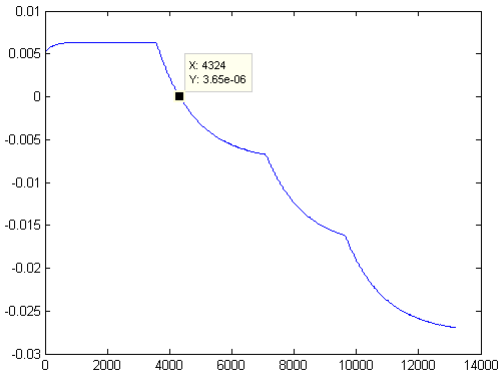


Figure 6.3: *Position on z-axis where phantom surface is found at z=0.*

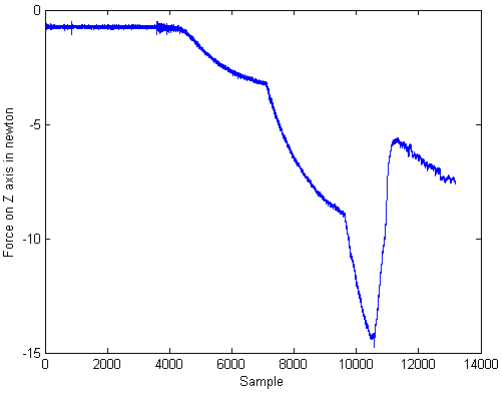


Figure 6.4: *Force on z-axis.*

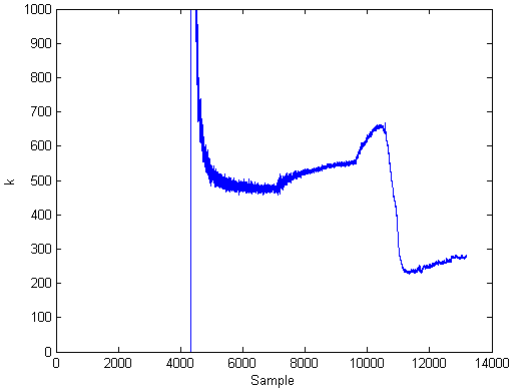


Figure 6.5: *K calculated.*

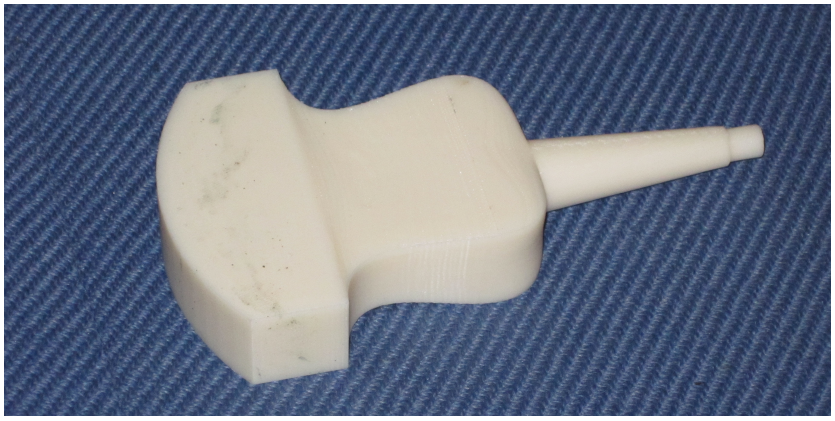


Figure 6.6: *3D-printed model of ultrasound probe.*

6.1 System-specific

In order to analyze the system response and capabilities, a few objective tests were done with the system.

6.1.1 Bode plot

In order to test the system response the robot was moved around in all axes with each of the proposed controllers with haptic feedback, as well as a controller without haptic feedback. This was done both for movement in free space and in contact with the phantom.

6.1.2 Computational time

The computional time of the two loops were timed by built-in timing functions in the C++ library.

6.2 User experiments.

Volunteers were asked to try out the teleoperation system. The task at hand was to use ultrasound to find a sphere inside the self made phantom (figure 6.7). Within the phantom, several objects are placed: Bolts, screws, electrical cables, a lego cube and two spheres. The same task was given for all the different controllers, as well for one without haptic feedback and one run where they used their own hands. The point of finding the sphere is not the real objective of the experiment, but simply a task to do that would mimic an ultrasound diagnostics session.

By performing this test we are checking how this system behaves seen from a users point of view while performing an ultrasound session. During the test it will be measured how accurately the test is performed and compared both between doing the ultrasound by hand and by teleoperation with and without haptic feedback. The goal of this experiment is to see whether it is possible to perform an ultrasound test

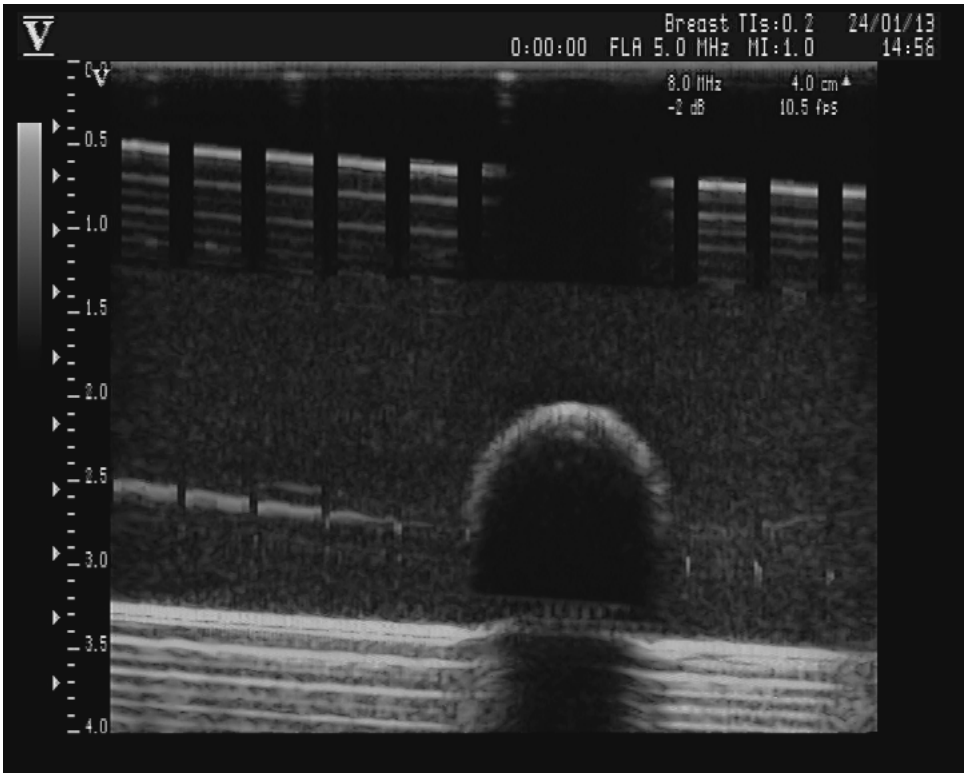


Figure 6.7: *Sphere in phantom as seen on ultrasound.*

with this system, and to see if there is any difference between having and not having haptic feedback. The things looked for in the experiment will be if there is a significant difference in the time usage, if the experience is better or worse with the different controllers seen from a user point of view. The excerted force is also important to measure in order to find out if a user excerts more force without the haptic feedback. After each session the subject were to fill out an evaluation form, this can be seen in the attachments section. The following questions were asked in the evaluation form (the ones marked with a star was not asked for the conventional ultrasound):

1.)	How was the overall feeling of the system. Unstable - Stable, 1-5	*
2.)	Was the response to your movement: Uncontrollable - Good, 1-5	*
3.)	How easy was it to anticipate the exerted force to the phantom: Not easy - Easy, 1-5	
4.)	Did the robot do what you wanted it to do? Yes / No	*
5.)	If no, please elaborate:	*

Table 6.1: *Questions asked for each controller*

The evaluation form can be seen in appendix C In addition to answering these questions, the performance of the users will be timed and their accuracy evaluated. By performing these tests it is checked how the system behaves seen from a users point of view while perfomring an ultrasound seesion. During the test it will be measured how accurately the test is performed and compared both between doing the ultrasound by hand and by teleoperation, as well as with and without haptic feedback. The goal of this experiment is to see if there is any difference between having and not having haptic feedback in a setup with ultrasound.

Chapter 7

Results

The system has been analyzed, and 5 subjects have performed user testing on the system. Some technical difficulties/problems was experienced during the user experiments. These are mentioned at the end of this chapter.

7.1 System specific experiments

The following section contains the experiments performed to obtain the system specifics.

7.1.1 Loop run time

As can be seen in figure 7.1 and 7.2 there are virtually no difference in the computational cost between the different controllers.

Haptics thread

The haptics thread handles the reading from/to the haptic device. It is running on 500 Hz, which means it should be run every 2 milliseconds. As can be seen in figure 7.1, it operates well within this limit.

Robot thread

The robot thread runs at 125 Hz, which means it should run every 8 milliseconds. This thread also handles the computation regarding setpoints to the slave. As can be seen in figure 7.2 most runs it is found to be within 7.9 to 8.06 milliseconds.

7.1.2 System response

Measuring the system response of this system has been done by manually attempting to recreate inputs of different frequencies. Preprogramming inputs would not yield correct results. In order to do this properly, a 3rd robot should have been attached to the master and programmed to steer

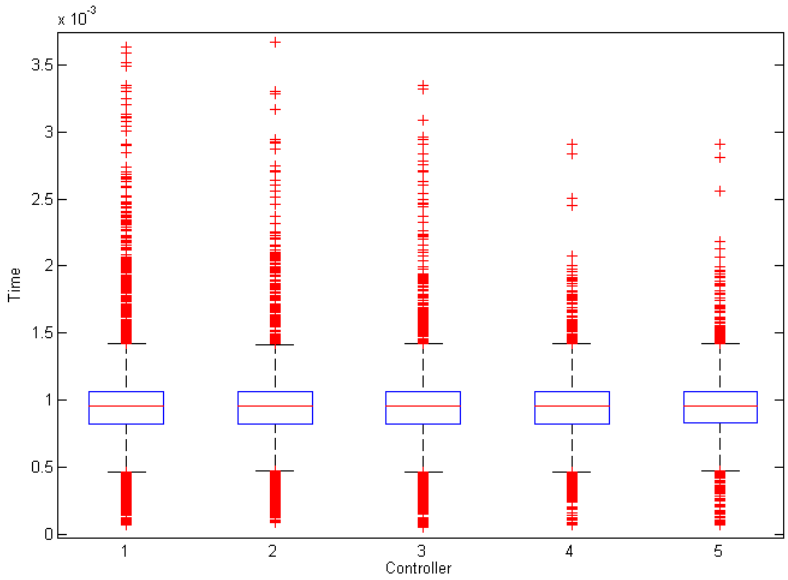


Figure 7.1: *Box and whiskers plot for the timing of the haptic control loop.*

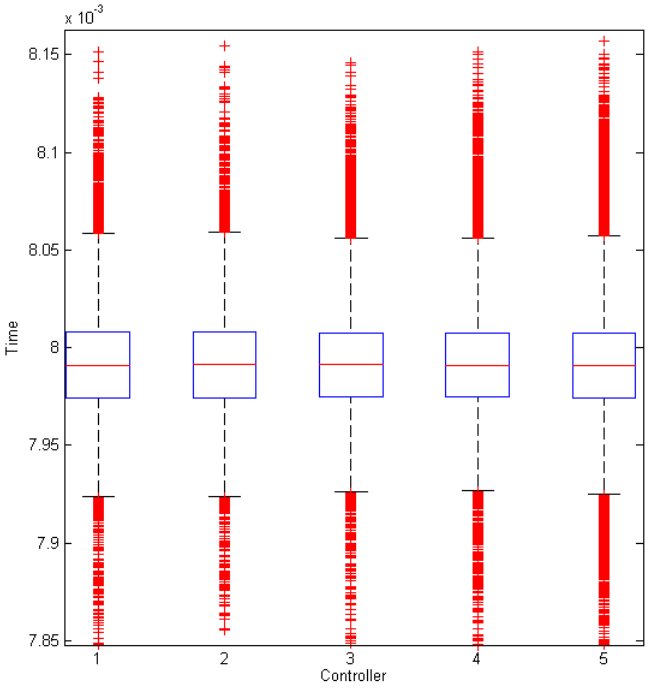


Figure 7.2: *Box and whiskers plot for the timing of the robot control loop.*

the master through the desired frequency range. This was however not available. The systems response is measured for movement in free air and for contact with the environment. For the contact with the environment several repeated contacts as well as contact over longer time periods has been recorded.

The bode plots are made in matlab where sampled data is modeled as a system in order to find a transfer function that fits the data. First, a system model is made by the *iddata()* method, then a state-space model is made by the *n4sid()* method. Finally the *bode()* method is applied to get the bode plot. In combination with the fact that data are not recorded through identical runs there is room for differences between the plots where they should have been the same. Findings for each controller will be commented. Following are a tables of the findings from the bode plots, and a few examples. Bode plots for all controllers are available in the appendix B.

The bandwith of a system is defined as where the dampening of the system is -3dB or less, and the frequency for which such dampening takes place is called the cutoff frequency. The phase shift is a measure of the delay in a signal, and when the delayshift has become -90 degrees or less, the signal is delayed by half a sample. These are the values to look for when investigating the system specifications. All values are read from the plots in appendix B by the help of the plot marker tool in matlab.

Velocity throughput

For the velocity throughput, derived speed from the Phantom Omni is used as input, and then the speed set on the UR5 robot as the output.

Linear velocity	Free space			In contact		
	0 Hz	-3 dB	±90° shift	0 Hz dB	-3 dB	±90° shift
Transparent	-0.15 dB	28.00 Hz	25.0 Hz	-0.19 dB	na	32.00 Hz
Static Low	-0.32 dB	na	36.6 Hz	-0.19 dB	na	na
Sigmoid Zero	-0.42 dB	2.70 Hz	12.4 Hz	-2.19 dB	5.2 Hz	10.00 Hz
Sigmoid Low	-0.24 dB	2.13 Hz	na	-1.30 dB	9.0 Hz	18.00 Hz
No Haptics	-0.10 dB	11.00 Hz	28.5 Hz	-0.08 dB	34.0 Hz	23.50 Hz
Angular velocity						
With haptics	-1.08 dB	0.85 Hz	1.75 Hz	-5.56 dB	na Hz	1.25 Hz
Without haptics	-0.80 dB	0.80 Hz	1.12 Hz	-4.80 dB	na Hz	1.20 Hz

Table 7.1: *Velocity throughput, system response*

In table 7.1 some cells contains "na" - not available. For these plots the amplitude gain either was already below -3dB or never went below, or their phase shift did not start on 0. For the velocitythroughput the initial dampening is just below 0 dB. The largest bandwith is found for the transparent controller in free air, at 28 Hz. The phase shift is found between 10 and 32 Hz for the linear velocities, and 1.12 - 1.75 for the

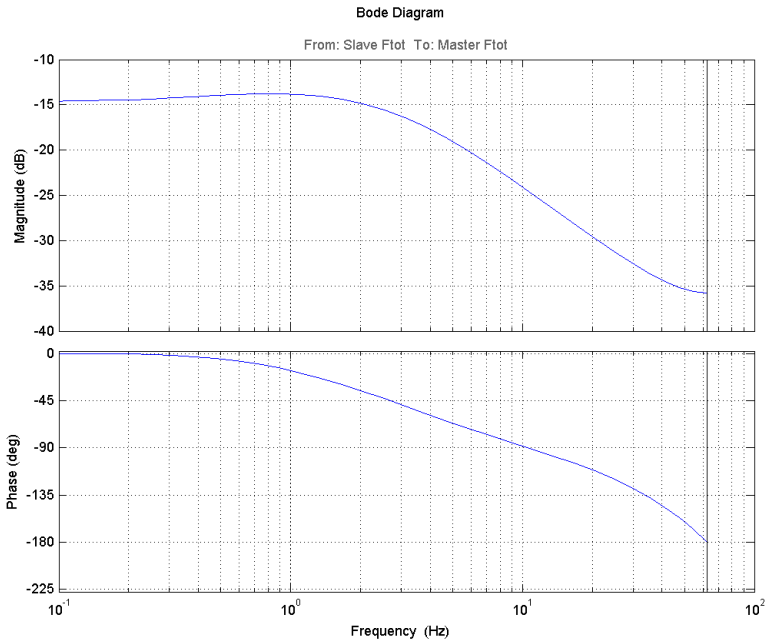


Figure 7.3: *Bode plot of force throughput in sigmoid zero controller in contact with phantom*

angular velocities. An example of a bode plot for velocity throughput and how to read it can be shown in figure 7.4.

Force throughput

For the force throughput measured force at the slave side is used as input and the force commands given to the Phantom Omni as output. Also, as all seems damped the cutoff frequency of -3 dB is never mentioned as this is allready passed. For the force througput only the case of in contact is analyzed as there is only supposed to be forces present in this case.

	In contact		
	0 Hz	-3 dB	−90° shift
Transparent	-12.3 dB	na	11.0 Hz
Static Low	-15.4 dB	na	10.0 Hz
Sigmoid Zero	-14.6 dB	na	11.0 Hz
Sigmoid Low	-16.6 dB	na	14.5 Hz
Without haptics	na	na	na

Table 7.2: *Force throughput, system response*

The force throughput is clearly damped,

7.3·

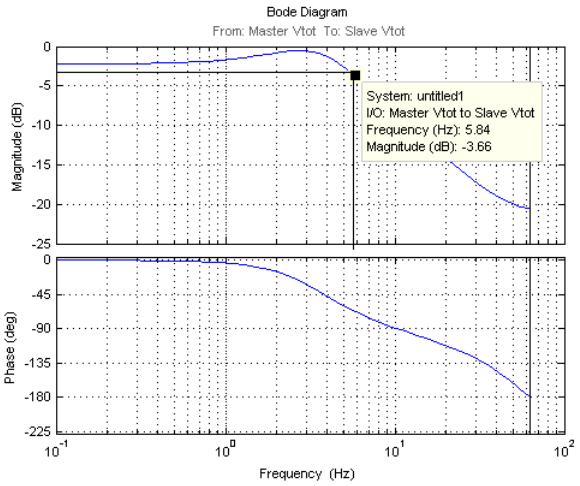


Figure 7.4: *Bode plot of velocity throughput in sigmoid zero controller in contact with phantom*

Transparent

Figure B.1 is the bode plot for the speed through the transparent controller moving in free space. This is damped by -3dB at 28 Hz, with the phase shift at 25 Hz.

In figure B.2 is the system response for the transparent controller while it is in contact with the phantom. It is initially damped by 12.5 dB with a phase shift at 11 Hz.

On figure B.3 the system response regarding the speed as the slave is in contact with our phantom is shown. The system gain is increasing as we are closing in to the half the sampling frequency. The phase shift seems to be rather equal to the run in free space.

Static low

Figure B.4 shows the bode plot for the speed of the static low controller. The gain increase when it closes in on the half the sampling frequency. The phase shift here though is very good all the way to haÃ¸f the the sampling frequency and is not falling significantly before that.

The force response in figure B.5 for contact with the phantom is again quite damped initially. The phase shift falls slightly steeper from 7-8 Hz.

In figure B.6 the system leads to a big instability with a large gain when frequency goes toward half the sampling frequency. The phase response on this is actually starting at -360 degrees, which means the signal is delayed by one cycle. it sinks slowly until it abruptly goes up to -180 degrees about the same time as the amplitude respons spikes past 15 dB.

Sigmoid Zero

The response to the speed seen in figure B.7 has a significant drop between 3 amd 7 Hz. The phase shift is sinking evenly between 0 and -180 degrees with a spike between 5 and 8 degrees.

In contact with the phantom, this controller has the force response of figure B.8. As all the other bode plots it has an initial damping of around .15 dB. Then it seems to be The phase shift evenly sinks to -180 degrees throughout the interval.

The velocity response in figure B.9 is initially damped with -2.19 dB and has an increase in gain around 3 Hz, but never goes over 0 dB. The phase response does not pass -90 degrees until 10 Hz.

Sigmoid Low

In figure B.10 we see the same pattern as in figure B.7, except that the dip is comeing already at 3 Hz. The phase shift however starts at 360 and goes all the way down to -180 with a very steep part around 10 Hz.

In contact with the phantom, figure B.11, the force is initially damped by about -16/-17 N with a small dip at 1 Hz. The fase shift also does not pass -90 degrees until 15 Hz.

When in contact with the phantom, figure B.12, has a very good amplitude and phase response. The magnitude is damped by 3 dB at around 34 Hz, and the phase is shifted by -90 degrees at approximitaly 23.5 Hz.

No haptic

For the controller without haptic feedback running in free space (figure B.13, there is some magnitude variations when passing 10 Hz. The phase also starts shifting at 10 dB.

When in contact with the phantom there is a large increase in the magnitude at abiut 25 Hz, before it goes down again to -3 dB at 34 Hz. The phase response goes to -90 at about 25 Hz as well.

Angular velocity - without haptic feedback

The angular velocity without haptic feedback has an initial dampening of -0.8 dB when running in free air. It is damped by -3 dB at 0.8 Hz, and the phaseshift of -90 degrees is to be found at 1.12 Hz. When contact is made with the phantom the system gets more damped, with an initial dampening of -4.80 dB. Thus the cutoff frequency of 3 dB is already passed. The phaseshift for -90 degrees is now at 1.2 Hz.

Angular velocity - haptic feedback

With haptic feedback, the initial dampening of the system is -1.08 dB for run in free air. It is damped by 3 dB at 0.85 Hz, and the phase shift to -90 degrees takes place at 1.75 Hz. For the contact with the phantom the

system was more damped, starting at -5.56 dB. Which again is passed -3 dB. The phase shift to -90 degrees at 1.25 Hz.

7.1.3 Stability

Some stability issues have been encountered when it comes to staying in contact with a surface. Figure 7.5, 7.6 and 7.7 is a sampled series of a contact with a rather stiff surface which leads a large force building up and pushing the master and operator back. It is clearly seen that the force increases quickly in only four samples as the slave comes in contact with the object. This instability is not so prominent for less stiff objects, like the phantom made for the experiments. Since the scope of the main project is a teleoperated semi-autnomous robot-system for use in ultrasound diagnostics, the use-case scenario does not involve objects with very high stiffness. Thus it has not prioritized to persue the stability for contact with high stiffness objects. In figures 7.8, 7.9 and 7.10 an example of contact with a soft surface, the phantom, can be seen. It can be seen that the raw force doesn not rise as quickly as for the stiffer surface. This is behaving differently, and the user is not pushed away.

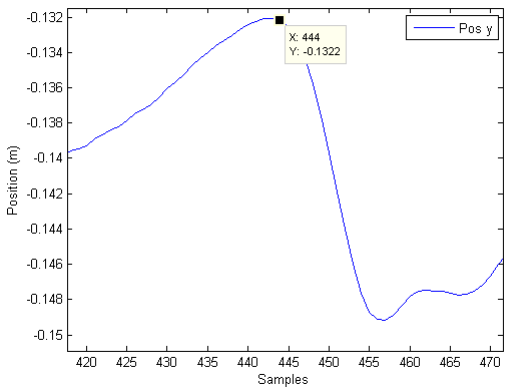


Figure 7.5: *Trajectory when pushed back due to instability as seen in the y-plane.*

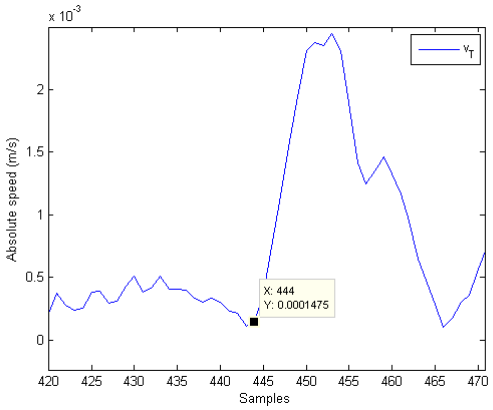


Figure 7.6: *Absolute speed when pushed back due to instability.*

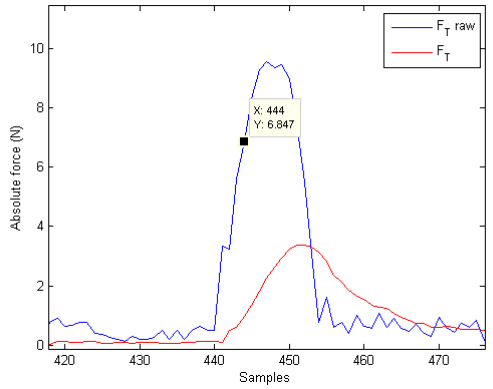


Figure 7.7: *Force when pushed back due instability.*

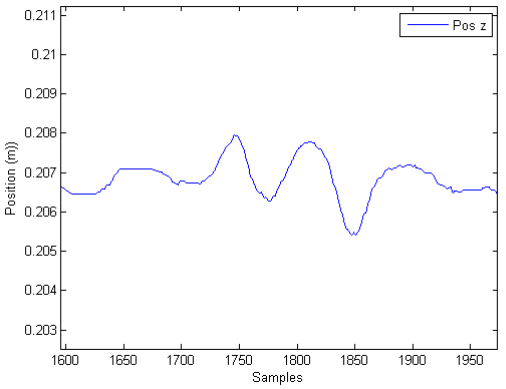


Figure 7.8: *Trajectory of contact with soft material in the z-plane.*

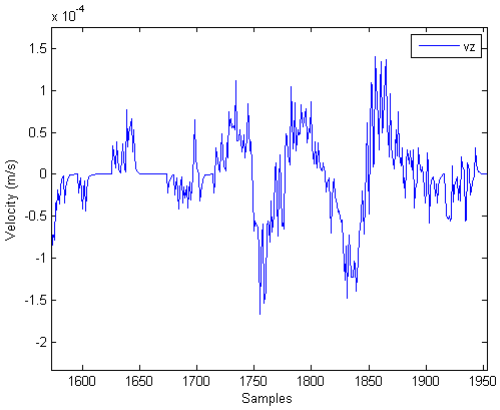


Figure 7.9: *Velocity of contact with soft material in the z-plane.*

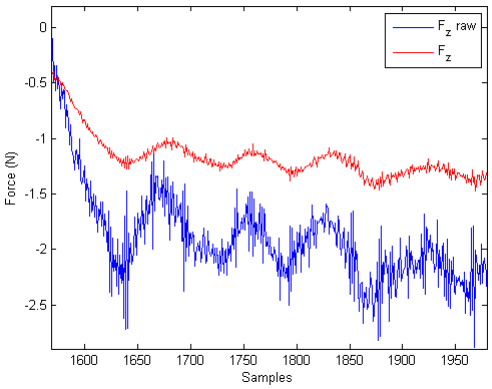


Figure 7.10: *Force from contact with soft material in the z-plane.*

7.2 User experiments.

5 subjects volunteered to try out the teleoperation system. The task at hand was to use ultrasound to find a sphere inside a self made phantom. Within the phantom, several objects were randomly placed. Bolts, screws, electrical cables, a lego cube and two spheres. The volunteers have the following qualities and experience regarding ultra sound on a scale from 1 to 6 where 1 signifies first time and 6 expert. I have also taken the test, in order to have a comparison with someone more experienced with the setup. The results from my run is not a part of the calculated averages, and are only displayed for comparison. The subjects only performed one repetition each for each of the controllers. Data on the participants is displayed below.

	Age	Occupation	US experience
Subject 1	28	PhD student, robotics	3
Subject 2	32	Researcher, robotics	4
Subject 3	27	Researcher, image analysis	1
Subject 4	34	Researcher	2
Subject 5	25	Researcher	1
Myself	27	Masterstudent	3

Table 7.3: *Participants in user experiments*

None of the subjects got any time to learn to use the teleoperating system, except for an explanation of which button to push in order to start the operation. The users all tested the controllers in the same order:

1.)	Conventional ultrasound
2.)	Transparent
3.)	Static low
4.)	Sigmoid zero
5.)	Sigmoid low
6.)	No haptic feedback

Table 7.4: *Controller testing order*

7.2.1 Objective User Evaluation

In the task-specific part of the experience, performance were timed and their accuracy evaluated according to page 4 on the attached evaluation form. This accuracy evaluation is ranged with 5 boxes from clumsy to spot on. In this scale, "clumsy" can be mapped to the value 1 and "spot on" the value of 5. The results from the experiments is lined up in tables, seperated from each controller.

The following tests have been done with the criteria as defined:

Test	Criterion definition
Task completion time	The time it took to find the sphere
Accuracy	Assessment of how well the ultrasound probe was maneuvered and evaluation of smoothness of movement.
Impact on environment	Measurement of forces exerted to the environment and Δp, depth deformation of phantom

Table 7.5: *Definition of tests and measurements for the objective user evaluation.*

Task completion time

	Mean	Median	Stdev	Maximum	Minimum
Conventional ultrasound	0:34:00	0:37	0:11:05	0:43	0:15
Transparent	3:01:40	3:45	1:50:67	5:01	0:57
Static Low	1:56:40	2:03	1:16:51	3:28	0:16
Sigmoid Zero	0:33:40	0:36	0:10:53	0:42	0:17
Sigmoid low	1:59:00	1:17	1:50:40	5:01	0:37
Without haptic feedback	1:21:20	0:51	1:03:87	3:12	0:37

Table 7.6: *Summary task completion time; finding sphere*

The time spent varies and the maximum and minimum time spent are distributed among the controllers. The sigmoid zero controller stands out as the same time is spent as without the teleoperation system.

Accuracy

	Mean	Median	Stdev	Maximum	Minimum
Conventional ultrasound	4	4	0.70	5	3
Transparent	2.6	2	1.34	4	1
Static Low	3.8	4	0.45	4	3
Sigmoid Zero	4.2	5	1.10	5	3
Sigmoid low	3.8	4	1.10	5	2
Without haptic feedback	3.4	4	0.90	4	2

Table 7.7: *Accuracy of performed task*

The accuracy performance from the sigmoid zero stands out again as beeing just as good as without the teleoperation. For the rest of the controllers, the accuracy is spread throughout the scale. The transparent controller is reported with the worst accuracy.

Impact on environment

The scalar force is calculated in order to find the orthogonal force exerted onto the environment. Then the samples where the slave is in contact with the phantom are identified, this is done by looking on the position of the slave. With these samples, features of the impact can be analyzed. All samples are taken account for, and the maximum, mean median and standard deviaton of the exerted force is found. The same is done for the depth. Regarding the depths, 0 is the surface of the phantom and positive values is within the phantom, e.g. Δp.

Force	Max	Mean	Median	Stdev
Conventional ulrasound	na	na	na	na
Transparent	36.08 N	2,78 N	2.17 N	2.22 N
Static Low	61.73 N	3.28 N	2.20 N	3.17 N
Sigmoid Zero	7.48 N	2.45 N	2.24 N	1.00 N
Sigmoid low	30.08 N	2.72 N	2.02 N	2.59 N
Without haptic feedback	15.61 N	3.50 N	3.09 N	1.53 N

Table 7.8: *Summary of force exertion on phantom*

It is found that the maximum force exerted is larger for the controllers with haptic feedback, except for the sigmoid zero controller. Looking at the mean and median, more and stronger forces are exerted by the controller without haptic feedback. Box and whiskers plot of the dataset can be seen in figure 7.12 and 7.14.

Depth	Max	Mean	Median	Stdev
Conventional ulrasound	na	na	na	na
Transparent	2.64 cm	0.33 cm	0.19 cm	0.30 cm
Static Low	2.55 cm	0.67 cm	0.74 cm	0.41 cm
Sigmoid Zero	0.93 cm	0.42 cm	0.32 cm	0.26 cm
Sigmoid low	2.69 cm	0.55 cm	0.41 cm	0.48 cm
Without haptic feedback	2.70 cm	1.34 cm	1.54 cm	0.61 cm

Table 7.9: *Summary of depth in deformation of phantom*

From the maximum depth it can be seen that all controllers except one went all the way down to the bottom of the phantom. Looking at the median and mean it can be seen that without haptic feedback in general the phantom ultrasound probe is pushed deeper into the phantom. The standard deviation is larger as well, indicating that there has been more movement without haptic feedback. Box and whiskers plot of the data set can be seen in figure 7.11.

Gain, k_v	Min	Mean	Median	Stdev.
Conventional ulrasound	na	na	na	na
Transparent	1.0000	1.0000	1.00	0.0000
Static Low	0.4000	0.9948	1.00	0.0428
Sigmoid Zero	0.0184	0.8533	1.00	0.1945
Sigmoid low	0.4075	0.7990	0.85	0.1727
Without haptic feedback	na	na	na	na

Table 7.10: *Summary of gain calculation*

The k is calculated for each axis in the implementation, but the results here are presented as the scalar of the 3-dimensional vector which in turn is normalized for easier interpretation:

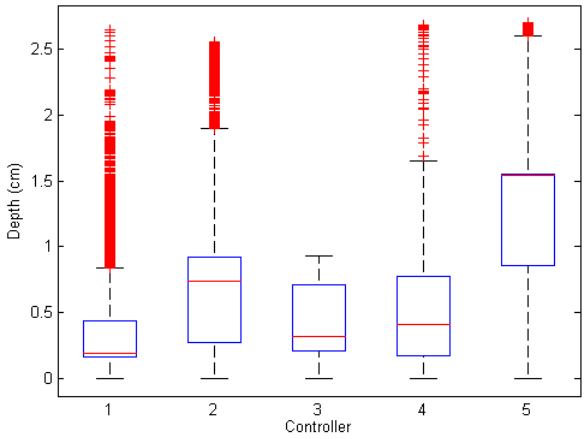


Figure 7.11: *Box and whiskers plot of the depth on the data for the different controllers.*

$$|k_{vNORM}| = \left\| \begin{bmatrix} k_{vx} \\ k_{vy} \\ k_{vz} \end{bmatrix} \right\|_{NORM} = \frac{\sqrt{k_{vx}^2 + k_{vy}^2 + k_{vz}^2}}{\sqrt{3}} \tag{7.1}$$

The sigmoid zero controller is the one that has dampened the velocity the most, while the sigmoid low seems to be the one that has been the most active.

7.2.2 Subjective User Evaluation

In the experiment, the user was asked to answer 5 questions about the experience according to the attached evaluation form, which can be found in appendix C. The statistics of the answers are calculated beneath, and a full answer table is attached in the appendix D. The definition of questions and answers is as follows:

Test Overall feeling	A subjective evaluation of the overall stability of the system
Response to movement	A subjective evaluation of the response to movement from uncontrollable to good
Finding the target	A subjective evaluation of the difficulty finding the target
Assessment of force exertion	A subjective value of the diffculty predicting the forces applied to the environment
Robot behaviour	A sbujective evaluation of how well the robot repeated desired movements
Elaboration	A possibility for the subjects to elaborate outside multiple choice

Table 7.11: *Definition of criteria for subjective user tests*

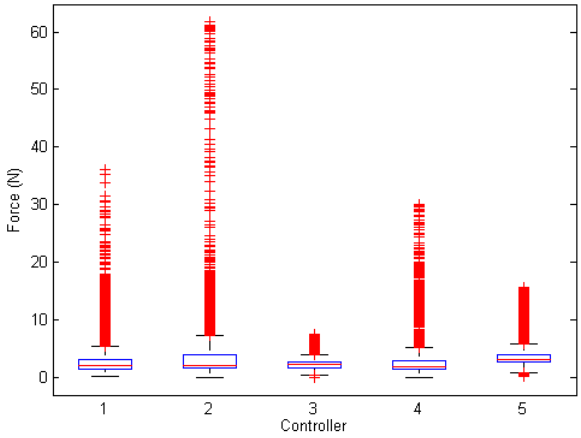


Figure 7.12: *Box and whiskers plot of the force on the data for the different controllers.*

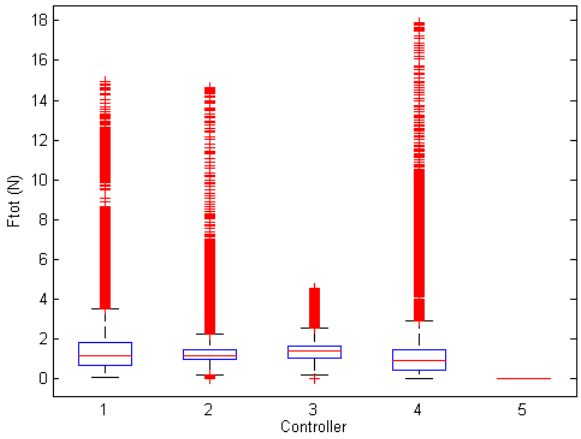


Figure 7.13: *Box and whiskers plot of the force to the master for the different controllers.*

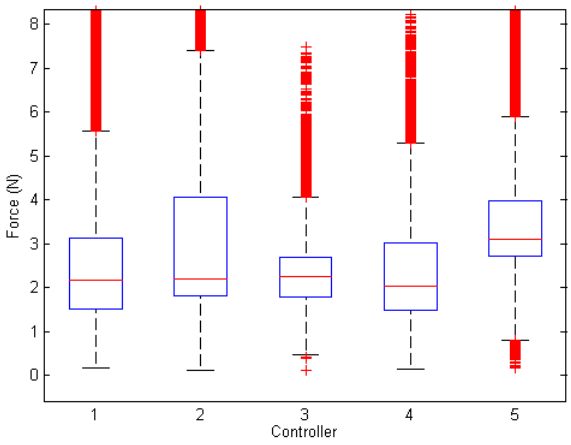


Figure 7.14: *Box and whiskers plot of the force on the data for the different controllers, zoomed in to match the whiskers.*

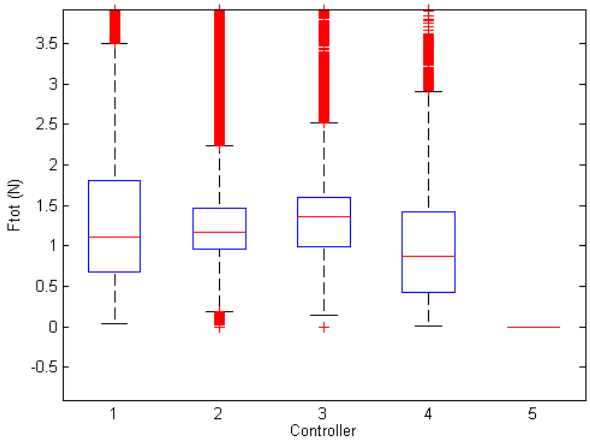


Figure 7.15: *Box and whiskers plot of the force to the slave for the different controllers, zoomed in to match the whiskers.*

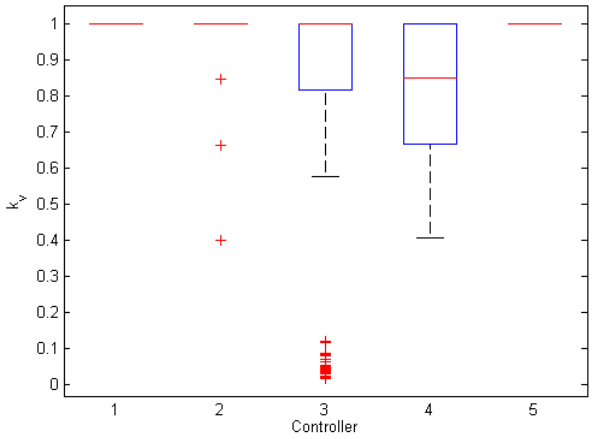


Figure 7.16: *Box and whiskers plot of the controller gain on the data for the different controllers.*

Overall feeling of the system

	Mean	Median	Stdev	Maximum	Minimum
Conventional ultrasound	na	na	na	na	na
Transparent	3.6	4	1.14	5	2
Static Low	3.8	4	0.84	5	3
Sigmoid Zero	4.4	4	1.10	5	4
Sigmoid low	4	4	1.10	4	4
Without haptic feedback	4.8	5	0.90	5	4

Table 7.12: *Summary overall feeling of the system*

The subjects are consistently reporting that they experience the system without haptic feedback as more stable than when haptic feedback is present.

Response to movement

	Mean	Median	Stdev	Maximum	Minimum
Conventional ultrasound	na	na	na	na	na
Transparent	3.4	3	1.14	5	2
Static Low	3.8	4	0.84	5	3
Sigmoid Zero	3.6	4	1.14	5	2
Sigmoid low	3.8	4	0.44	4	3
Without haptic feedback	5	5	0.00	5	5

Table 7.13: *Summary of response to movement*

Without the haptic feedback, the response to movement is reported to be good from all subjects. For the controllers with haptic feedback the reports are mostly good too.

Finding the target

	Mean	Median	Stdev	Maximum	Minimum
Conventional ultrasound	4	4	0.71	5	3
Transparent	2.8	3	1.30	4	1
Static Low	3.2	3	1.10	5	2
Sigmoid Zero	3.8	3	1.10	5	3
Sigmoid low	3.6	4	0.55	4	3
Without haptic feedback	4.2	5	1.30	5	2

Table 7.14: *Summary of finding the target evaluation*

The users found it just as easy to find the target with the teleoperation as without. However with haptic feedback they report it to be less easy.

Anticipation of exerted force

	Mean	Median	Stdev	Maximum	Minimum
Conventional ultrasound	4	4	1.15	5	3
Transparent	3.2	3	0.83	4	2
Static Low	3.6	4	1.14	5	2
Sigmoid Zero	3.6	3	0.89	5	3
Sigmoid low	3.8	4	0.84	5	3
Without haptic feedback	2.6	2	1.81	5	1

Table 7.15: *Summary, anticipation of exerted force*

It is reported to be much harder to anticipate how much force is exerted onto the environment without the haptic feedback. Also, it can be seen that the subjects reports it as easier to anticipate for the last controllers with haptic feedback than the first.

Robot behaviour

"Did the robot do what you wanted it to?"
Here the possible answers are yes and no. For statistical purposes the answer yes = 1 and no = 0.

	Mean	Median	Stdev	Maximum	Minimum
Conventional ultrasound	na	na	na	na	na
Transparent	0.6	1	0.55	1	0.0
Static Low	0.5	0.5	0.50	1	0.0
Sigmoid Zero	0.9	1	0.22	1	0.5
Sigmoid low	0.9	1	0.22	1	0.5
Without haptic feedback	1.0	1	0.00	1	1.0

Table 7.16: *Summary of robot behaviour*

Subjects reported that they felt the robot mostly did what they wanted it to. The ones that are answered no can be traced back to some problems with the robot.

Elaboration

Subjects elaborated a bit on the following controllers:

	S1	S2	S3	S4	S5	Myself
Conventional ultrasound	na	na	na	na	na	na
Transparent	1.)	na	na	6.)	10.)	na
Static Low	2.)	na	5.)	7.)	11.)	na
Sigmoid Zero	3.)	na	na	8.)	na	na
Sigmoid low	4.)	na	na	na	na	na
Without haptic feedback	na	na	na	9.)	na	na

Table 7.17: *Summary of elaboration*

	Controller	Elaboration
1.)	Transparent	Hard to keep the contact and get the probe orthogonal to the phantom.
2.)	Static Low	A bit the same as number 1.
3.)	Sigmoid Zero	The same as 2.
4.)	Sigmoid Low	The same as 2.
5.)	Static Low	At the ends it was stopping abruptly.
6.)	Transparent	Yaw a bit difficult.
7.)	Static Low	Bumping up and down.
8.)	Sigmoid Zero	Difficult to control angles.
9.)	Without haptic feedback	Hard not to crush the patient.
10.)	Transparent	It is like the coordinate system changed.
11.)	Static Low	Better than before, but still hard to control.

Table 7.18: *Elaboration*

7.2.3 Problems and abruptions during experiments

For subject number 2 it was necessary to do a re-estimation of the payload when doing the test with the sigmoid low. After the robot has been on for a while it gets a bit warm in the joints. Since the force sensor is attached to the last joint, it is affected by this and get's warmed up as well. As the force sensor is warmed up it's mechanical properties change; which in turn causes the readings to change.

For subject number 4 the robot went into security stop during the run without haptic feedback. Security stop is a security function in the robot that prevents the robot from doing something harmfull. In this case, there is a malfunction in some firmware that stops joint one. This happens often when this joints is rotated past a certain angle, and then it needs help to get up again. A restart helps too, but often takes more time. During the sigmoid zero run, it was also necessary to do a re-estimation of the payload for this subject.

A re-estimation of the payload was also necessary to do during the transparent run for subject 5.

Chapter 8

Discussion

The results have been presented, and they will now be discussed. In section 8.1 the system-specific results will be discussed; how well have the system performed? Has it performed according to expectations? In section 8.2 the results from the user experiments are discussed and the lines towards the conclusion are drawn. Then finally in section 8.5 the limits and weaknesses of this setup are discussed.

8.1 System specific experiments

In this section the general performance of the system is discussed. Loop run time, system response and stability will be discussed.

8.1.1 Loop run time

The time the algorithm runs in the robot thread is at the edge of what is possible within the time-slots available, and as seen in the results it sometimes uses a few microseconds more than it has got available. In the computation of the setpoint there are lots of heavy linear algebra calculations. The inverse kinematics, the forward kinematics, the inverting and the multiplication of several rotation matrices. In addition data is beeing logged to the disk, which is a process that takes up a lot of time. Since there are two threads operating on the same struct in this program, there are also locks implemented to lock certain common variables as one thread uses it. This way the threads might have to stand in line and wait for variables to be available. By operating in angular and cartesian velocities the computation time of the slave thread would probably have gone down. If aquiring a new haptic device, this is something to bear in mind in order to optimise the run-time. The haptic thread however is working well within it's limits, and even have some time to spare.

8.1.2 System response

The system response have been analyzed, and it has been found that the force throughput are damped quite a lot. If the force read from the sensor

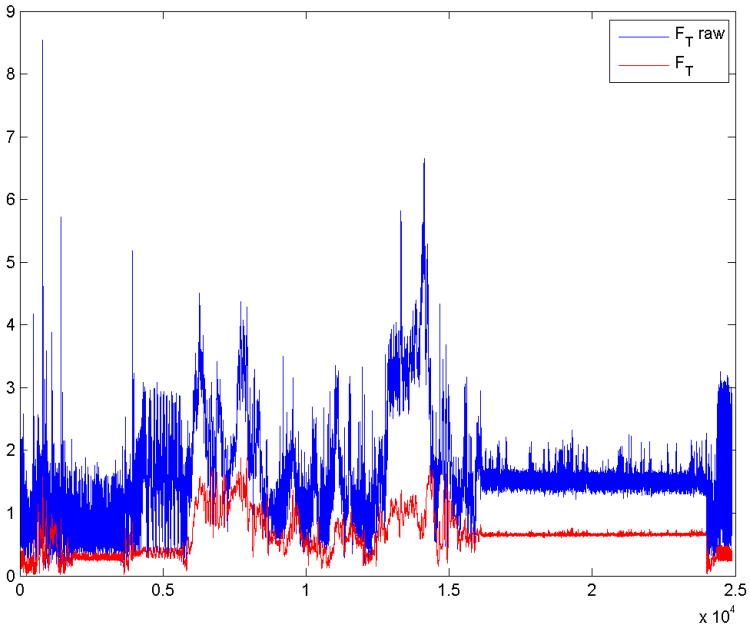


Figure 8.1: *Example of raw and processed force plotted in the same plot.*

is plotted along with the this can be seen visually as well, an example of this can be seen in figure 8.1. This means that the force is not transparent in our system. As can be seen in the figure, the noise is quite high at times. Considering that the Phantom Omni is saturated at 3.3 N the haptic feedback would not of any good use as it would be hard to know if you were feeling the environment or the noise. Also, having noise at that amplitude, which was experienced before any filtering was implemented, was quite useless as it is hard to hold the Phantom Omni still with high-frequent force like that. That way the noise gained amplitude through the system loop and rendered the entire system quite unstable. Thus allthough it was desired to have as transparent force throughput as possible, it was necessary to apply much filtration on the force signal. As a result of the chosen filtering, the force was dampened as we see in the figure. This is most likely due to the two moving average filters that are applied in series. Besides the dampening, the force-response is good for low frequencies, which is what we need for an ultrasound examination system.

Concerning the velocity response it is initially slightly dampened, which can probably be explained by inertia in the system. For low frequencies the velocity response is very good, with a phase response. This is recognisable as the robot, when moving in free space, follows every movement in good fashion. This was confirmed by the subjects in the user test performed. For some of the bode plots, amplification is shown for high frequencies, however these are higher frequencies than is likely to be encountered in a clinical setup. Some plots also show a phase response starting at -360 and +180 degrees, which is found to be a bit odd. A

possible explanation for this can be the way these are calculated.

A system model is built in matlab based on the input and output data, e.g. master velocity and slave velocity. System modelling based on data is rarely accurate, and there might be some features regarding the dataset that makes system model inaccurate in these cases. Different parameters in the system model was tried out, with better phaseresponse reported for these plots, but overall the results in the bode plots was worse, thus the default settings was used. Another reason why these system models can be inaccurate is due to the way the data was collected. As stated in the results-session, data was collected by manually exciting the system in an attempt to go through all frequencies. Needless to say it is hard, if not impossible for a human arm to reach high frequencies like 10-30 Hz, so a lot of data is probably missing in order to make a more correct system model.

For the angular velocities, especially low values were found in the system response. If it is hard to reproduce a frequency range manually by exciting the system manually for linear movements, it is even harder for angular velocities. These values are not believed to be the full descripton of the system response for angular velocities. However the magnitude response is near 0 for low frequecies, which means that this system has a good response to our use-case. The corresponds well with experience from using the teleoperating system.

8.1.3 Stability

The system is mostly stable, except from at the moment of contact. During development it was experimented a bit with the global force scaling. As the force was scaled down, the stability upon contact became more stable. The phantom omni is quickly saturated forcewise upon contact since it cannot handle more than 3.3 N. This might be one of the reasons for this experienced instability, as the system then is pushed to the extremitys of what it can handle. As could be seen, contact with the softer surface did not result in the same sudden forces, which leads to the better contact. Still some minor vibrations / oscillations can be seen in figures 7.8, 7.9 and 7.10, but looking at the scale of the axis these are very small and can be neglected.

8.2 User experiments

The user experiments consisted of two simultaneous measurements of the system. One objective measurement, which looked on how the performance of the subjects was during the experiments. This is discussed in subsection . The other part was seeing how the users felt using the proposed system, which for a setup like this is just as important.

8.2.1 Objective user evaluation

5 different users tried out the master-slave system. Most of them had never tried performing ultrasound diagnostisation before. Only one of them had previously tried the master-slave system, allthough not by much.

Task completion time

None of the subjects had much experience with ultrasound examination on beforehand. Looking at table 7.6 we can see that they used all from 15 to 43 seconds to find the sphere using conventional ultrasound. With the teleoperating system more time was used. However, looking at the time and comparing to the order of which the controllers were tested it can be observed that the time goes down as the subjects gets more trials on the teleoperator. Considering that none of the subjects had tried the system before, this indicates that there is a learning involved.

For the *transparent* controller both the mean and the median is above 3 minutes, and even the minimum time is the highest minimum time of all controllers. Since the dataset contains five samples, the median confirms that more than half of the subjects spent more time than 3:45. This can be explained by the fact this was the subjects' first run with the teleoperator. Since they had never tried this before, they needed this run to familiarize themselves with the system and how it worked. The maximum time relates back to the technical difficulties experienced for one of the subjects in this run, which contributes to an increased mean time. Seeing that the mean is located 45 seconds below the median indicates that the two last samples are substantially lower than the rest. Seeing how the task of finding the sphere involves a bit of luck, it is not surprising to have a few trials where the sphere was found much faster.

Less time has been spent using the *static low* controller than the transparent one. The minimum time has even gone down by just as much as for the run without the teleoperation. Both the mean and median decreased by more than a minute, It seems the subjects have started to learn how the system reacts to their movements. This was their second turn with the teleoperator. It can be seen that the mean and median are much closer for this run, suggesting that the sampled time is more evenly distributed.

The *sigmoid zero* seems to be the most succesful run of all. Except for the minimum run, all of it's sampled values are well below even the conventional ultrasound. As previously stated, in this task there is an element of luck regarding the starting conditions of the search for the sphere. Different starting point or direction in the search might make a subject find the sphere after just a few seconds. However it is not likely that has happened to all 5 subjects at once. Also no technical problems was encountered during this run. This has of course contributed positively to this run, and shows that when everything works properly there is not nessecary any difference between using this setup compared to the conventional way. At least not for inexperienced people.

Looking at the *sigmoid low* a technical problem was experienced here for one of the subjects which makes the maximum time large. This in turn affects the mean, however we can see by the mean that the are lower than for the static low controller. If it hadn't been for the problem here the time values for this controller had probably ended up between the static low and sigmoid zero controller, as can be seen in table D.2.

Without the haptic feedback less time is used than for most controllers, except for the sigmoid zero. There were som technical difficulties during one of these runs which means that the mean could probably have been lower than it is. This was the subjects' fifth run with the robot, which means that they had gained som experience as to how the system reacts and how it is to operate. Even so, for all the runs more time was spent during this run than for the sigmoid zero controller.

After seeing how much time was spent finding the sphere with each of the controller it can be seen from the data that the users are gaining experience in how to use the system which makes them handle it better. It can also be seen from the sigmoid zero controller run that when the system is working properly, it does not necessarily increase the time spent to find something on an ultrasound examination. Due to the nature of the data it can not be concluded that less time is spent with haptic feedback than without, however the sigmoid zero controller is a clear indication in that direction. By the run without haptic feedback the subjects had more experience with the setup than on the sigmoid zero run. If there were no difference between the two it would be expected that the results from the two runs was more alike.

Accuracy

The accuracy was judged by how well the subjects handled the master-slave system from an observers point of view. How did they hit the surface, did they manage to follow the surface around, did they manage to get the correct orientation and general handling. The accuracy results (table 7.7) corresponds well with the timeing results (table 7.6). Not surpriceingly it can be seen that the sigmoid zero controller has the highest score of all, and the transparent controller the worst. However an interesting observation is that the controller with second lowest score is the one without haptic feedback. Without haptic feedback it is hard to know when you are in contact with the surface, which makes it hard to keep the probe accurately in place at the surface level.

Impact on environment

Before the experiments it was expected that the strongest exerted force to the environment would come from the controller without haptic feedback. Looking at the reported data in table 7.8, we see that the maximum force has not been exerted by the no haptic run. However if we look at the mean and median force we can see that stronger and more force have been exerted on the phantom without haptic feedback. If we look at the depth in

table 7.9 , e.g. how deep into the phantom we see the same trend. The higher mean suggests that more force have been exerted, and the higher median suggests that at least half of the sampled values have been higher than that. The reason why the maximum force is higher for the ones with haptic feedback is probably due to the technical problems some of the subjects experienced during the experiments. This led to the slave going all the way down to the bottom of the phantom exerting large forces up to 60 N and a depth of 2.55 cm. If it had not been for these incidents, the results without haptic feedback would probably have stood more out. A good indicator of this is the sigmoid zero run, which for all subjects went very well. It seems that the haptic feedback is of good help to the operator when it comes to estimating the force exerted to the environment and when it is stable it really helps in exerting as little force as possible.

Looking back on the visualization of the statistical data (figure 7.11, 7.12 and 7.14) we can also conclude that the force feedback is helping the operator in keeping the slave steady at it's target. The spread of the depth is much larger for the controller without haptic feedback than the ones without. Taking a closer look on the sigmoid zero controller it is also viable to say that the force exerted does not vary as much. For the other controllers with haptic feedback, we can see that they have just as large variation as for the controller without haptic feedback. However due the tehnical difficulties that produces some extreme values that affects our dataset statistically. If these extremeties were to be corrected for, the variation would probably be much less. If the plots between the master and slave are compared (figure 7.12 versus 7.13 and figure 7.14 versus 7.15) it is clear that the forces in the master corresponds well with the forces from the slave; only scaled down.

Gain computation

In figure 7.16 the dataset of the computed gains for k_v is shown. For the static low controller three levels of dampening can be seen. This is due to calculation as explained in equation 7.1. Seeing as this was come in effect when $|f| > |N_{MAX}|$ this has apparently not exceeded this limit a lot. This was implemented from the filtered force, and looking at figure 7.15 it is clear that this limit was rarely exceeded. In other word this has in practice been the transparent controller. The filtered force was applied as the limiter because of all the noise in the raw force from the slave. Had the noise from the force sensor been less it could probably be a good idea to have this limit be read from the slave side insted of the filtered master side.

The sigmoid controllers has been more active. Since the sigmoid zero controller have the possibility to converge to 0 when f becomes high, it was expected that this would have a bigger variation in it's value than the sigmoid low. However looking at the plot of it is the opposite way, apart from a few outliers near 0.1-0.0 for the sigmoid zero. This is consisten with both figure 7.14 and table 7.6. Seeing as much more time was spent during the sigmoid low run, there are much more data here which can lead to a bigger variation in the sampled data. Also, most importantly, more

force has been exerted for the sigmoid low run. As the k_v for the sigmoid controllers are directly related to the force exertion it should come as no surprise that the sigmoid low has a larger spread. Due to some of the technical difficulties experienced during the sigmoid low run the dataset contains some extreme values.

Other observations

When it comes to the relativeness of the orientation, this worked fine. However some issues regarding rotation around the yaw was observed. When the subkjects started the experiments, the ultrasound probe was aligned so that the image on the screen corresponded to the orientation of the ultra-sound probe. However after a while of searching, some of the subjects had turned this 180 degrees around the z-axis of the base coordinate frame. Then the image on the screen got mirrored compared to the orientation and movement of the of the robot with the proposed, which could be a bit confusing. When redesigning or optimizing this system, this could be something to look into for better usability. Either it could be possible to mirror the ultrasound image when the ultrasound probe passes a certain orientation, or one could implement to controller to have all rotations except for the yaw relative.

8.2.2 Subjective user evaluation

After each run the subjects had to fill out an evaluation form, which can be seen in appendix C. The results from these forms will be discussed in this section.

Overall feeling of the system

In table 7.12 it can be seen that the average reported feeling of the system is rising as the subjects gain more experience, with the best feeling beeing reported from the run without haptic feedback. Without haptic feedback there is no noise or unwanted force bias from the cables of the attached equipment to degrade the experience. The data is corresponding with the time spent, indicating that the subjects experience the system as overall good as they succeed well when performing the task.

Response to movement

In table 7.13 the run without haptic feedback gets top score. All subjects report it to be good. Of the controllers with haptic feedback, the sigmoid low and static low are reported to be the better ones. However the mean values are quite simular, indicating that there is not much differense in the reportings.

Finding the target

Looking at table 7.14, the subjects have reported it just as easy to find the target with the conventional ultrasound as with the teleoperator without haptic feedback. With haptic feedback the reported level of easyness is reported from 1 to 5. Considering the mean it can be seen that reported easyness is increasing for each run with the teleoperator, contributing to the assumption that there is a learning curve within these numbers. The sigmoid zero is again rated best of the controllers with haptic feedback.

Anticipation of exerted force

When it comes to the anticipation of exerted forces, the outcome is just as expected. It is reported to be easiest with the conventional ultrasound run. Then looking at the mean again the values is increasing for each run with the teleoperator (table 7.15). Since there is no difference between the controllers when it comes to the force feedback, other than the force bias from the cables and payload estimation that might differ, this outcome is quite as expected. For the transparent controller it is the first time they tried the system, and thus they had to start learing how to comprehend and adjust to the feedback. This can be seen to become easier for each run they have. For the controller without haptic feedback not surprisingly it is reported to be the most difficult one to anticipate force from. However one subject reported it to be easy. In order to assess this better it could be interesting to have them guess how much force was being exerted and compare this with actual measurements.

Robot behaviour

Again there is an increase in the mean value observed, looking at table 7.16. This is consisten with the already established learning effect within the data. Some of the subjects have reported that the robot did not do what they wanted it to at all. As the force biased from the cables have been different from subject to subject it might be that they experienced more powerful bias than the others. However if table D.8 in the appendix is consulted it can be observed that these answers are given for the first two runs with the teleoperator. This indicates that they might have experienced the system differently as they learned how to handle it. Subject one answered both yes and no on most of these questions.

Elaboration

At the end of the evaluation form the subjects could elaborate if they had a comment that was not covered by the multiple choice, see table 7.17 and 7.18. The two controllers most elaborated about is the transparent and the static low controller. Considering these were the two first runs with the teleoperator, it might be that part of the reason they elaborated was due to the fact that they were still learning to use the system. Subject one reports that it was a bit hard to get the ultrasound probe orthogonal to the

phantom and keep the contact. Of all the subjects, this was one of the fastest and most accurate subjects. Seeing as this was one of the subjects with most experience regarding ultrasound, this might cause this subject to see some issues that the rest of the subjects did not see.

Some difficulties were elaborated regarding orientation and rotation. This concurs with the observation of the subjects as they performed the tasks. As stated in the section 5.2.3 the slave base coordinate system was chosen as the common coordinate system. Thus the relative rotation in the master would be replicated in the base system of the slave. These implications of this is discussed further in section 8.5.7.

8.3 Overall findings

From the discussed results it can seem that the sigmoid zero controller is the better one out of the proposed controllers. It stands out as the controller where the users spent least time finding the sphere, as well as exerting less force and going less deep into the phantom. Although it is not possible to conclude that this is a fact based on the data gathered, it is a clear indication this kind of controller might be something to investigate further. If this is compared with the results found in the system specific experiments, especially table 7.1, it can be seen that of all the controllers, the sigmoid zero seems to be the most damped one with the lowest bandwith for the contact case. This indicates that this controller has been more damped than the others. This may have led to the increased performance for this controller amongst the subjects.

When it comes to haptic feedback versus no haptic feedback, it is a clear indication from the data that the presence of haptic feedback reduces the exerted force as well as the deformation of the phantom. Looking at the deformation, e.g. depth into the phantom it is seen that the variation is much less with haptic feedback than without, meaning the subjects have been able to keep the ultrasound probe at the approximate same depth into the phantom. It is easier to keep something steady for a human being when pushing towards an opposing force than when holding something in "thin air". This really speaks for having haptic feedback in such a clinical setup as it seems to be of good assistance to a user.

There is a clear learning curve in the results. Apart from the sigmoid zero controller that really stands out the results are consistantly better in the last run than in the first run. In order to better assess which of these controllers a blind trial should be hold where the order of the controllers were randomized, so that the results would not be biased from the order it was presented. Another way to avoid this would have been to give each subject some time and a different task to perform that would not have been measured so that the subject could familiarize itself with the system setup on beforehand.

8.4 Experimental weaknesses

There are some apparent weaknesses about these experiments. First of all, none of the participants in the user-study was actually a radiologist. This means that none of them really have good experience in doing ultra-sound diagnostics, and does not really feel the difference or pro et contra regarding this new setup compared to how it is done on patients. However, if unexperienced people are able to perform the exercises it is probable that it is possible for professionals as well. The fact that all of them are computer scientists suggests that they might be biased towards technology the that they are more positive towards it than a radiologist would have been. However being technologically comptenet they might see problems / challenges with the setup that a radiologist would't have.

Since the selection of subjects is small, the dataset is not statistically significant. In the user test they only had one run for each controller, which means that we only have a selection of 5 sets of data pr. controller. During the experiments, it would probably have been wiser to have the users doing repetitons of the tasks several times. Not only would you get a better dataset which would not be as vulnerable to extremeties as ours is today, but you would probably also get the benefits of the subjects learning to use the system better. After all, in a clinical setup this system would be used by experienced operators, meaning that the effect of first time use is not that interesting to this project.

8.5 System limitations

This system has been implemented using a Universal Robot UR5 and a Sensable Phantom Omni haptic device. The reason why these devices have been used is simply because these were the devices at hand. There seems to be a mismatch between these two devices. It is not necessary, nor desireable, to have the same scale of master and slave, at least in most cases. However by experience gained in this project the Phantom Omni is to small for this kind of setup.

8.5.1 System Modeling

As stated in the implementation section there has been no modelling of the master nor the slave. It is assumed that this takes place at the device driver level. The obvious advantage in this is that you get completely independent of what device you have implemented either as a slave or a master. It get's quite easy to change these when necessary. On the other hand, this might be the reason why there are som stability issues when it comes to the moment of surface contact. This is something to look into, if it is possible to make a generic spring damper mass model that can kick in right in that moment. Perhaps by using external measurements as well, like 3D tracking [27], it would be possible to identify when the master is getting close to the environment and thus change the controller the way i

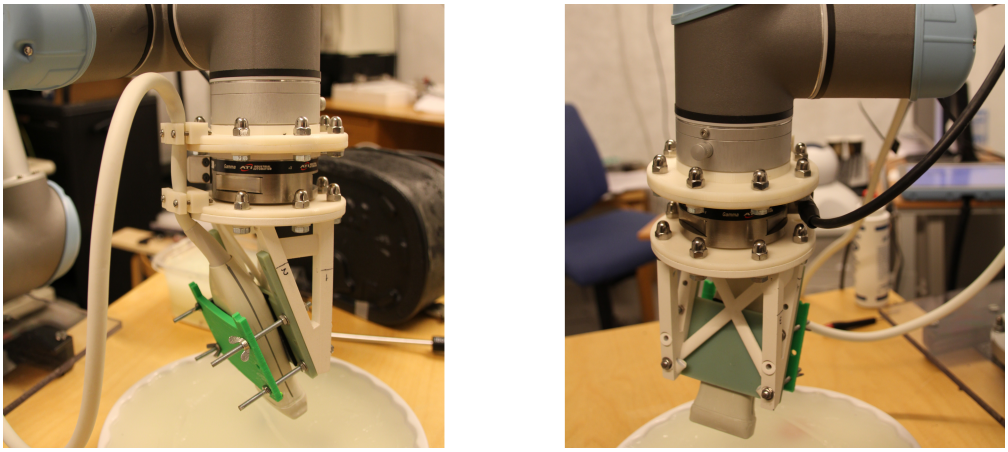


Figure 8.2: *Ultrasound and force sensor cable on the robot.*

have proposed in this thesis. A combination of these two might be to prefere, as you would then be able to prepare the robot for contact with the environment based on the visual feedback and then confirm it based on the force feedback.

8.5.2 Payload estimation / force bias

In order to measure the impact on the environment of the robot precisely, the payload attached to the tool-frame of the robot needs to be estimated in order to apply a force cancelling out the force applied to the robot by the payload. The force-sensor and the ultra-sound probe are both attached to the robot with cables hanging freely from both of them (figure 8.2). In early testing of the system, the cable from the force-sensor has never been a big issue. When the ultra-sound probe was attached, this is a thicker and much heavier cable, the drag from the cable was suddenly noticed as a force bias for certain orientations. As the robot moves around the mass center of the payload changes, as the cable changes position. This is rather hard to estimate, and therefore it is experienced a force offset when there shouldn't be one as the ultra-sound probe is attached.

8.5.3 Ergonomics

The ergonomics of the Phantom Omni, is not optimal for this use-case scenario. The Phantom Omni end-effector is formed like a pen. This is ideal for use like 3D-modelling and manipulation of virtual environments from a desktop computer. For a radiologist performing ultra-sound diagnostics, the grip you have when holding a pen will not be good for maintaining a certain force for a prolonged period of time. Today, radiologists performing ultra-sound is having a firm grip with the entire fist in order to have good control of the force exertion and the position of the ultra-sound probe. Now, one could argue that with the force-scaling involved it could be that it is not necessary to have the firm grip that they

use today, but it might be sufficient with a pen like grip. Some day in the future that might be, with the right control algorithm, but for now it would not be like that. Radiologists are used to work a certain way, and for them to take a system like this into use, it would have to resemble their old ways of working in order for them to want to take it into use. We humans are performing best within the comfort zone of our own habits.

8.5.4 Mechanical difference

Another thing is the strength of the Phantom Omni. It is only guaranteed to be exerting 3.3 N [1]. This is nowhere near what the UR5 is capable of exerting. Also it is only half of what are beeing used in ultra-sound sessions [51]. One thing that has been noticed while working with the Phantom Omni is that 3.3 N is not enough to stop a user from going on. It is possible to push on without the force getting stronger, which means that the robot will follow me down and the force exerted to the environment will increase. Some times, if the force would be too much for the phantom, it would simply "give up". This is probably some kind of built-in safety mechanism in order not to overload the actuators, but the effect is that when it "gives up" the opposing force dissappears while the operator is still exerting his/her force. That way the operator will steer the master fast down past the point where he was, and the slave will go after down into the environment. At this moment force feedback is turned off in the Phantom Omni so the operator will not know how much force is beeing applied to the environment, which could be critical if this was a patient.

If we look back to the extended lawrence architecture, figure 5.3, we see that the impedance of the master and the slave is a part of the loop. While the UR5 is large manipulator with metal couplings, strong joint motors and large mass compared to the Phantom Omni. This means that if the master hits or collides with something, it doesn't take nearly as much force to stop the movement of this than it would if it was the slave. When passing the force on from slave to master without scaling the force down, the result will be a sudden large force on the impact, as seen in section 5. One way to deal with this is to to scale the force down. The downside to scaleing the force down is that you lose some of the sense of the environment, however the question remains: Is it necessary to have full force transparency, or is it enough to have the force scaled down a certain amount in order to cope with the difference in mechanical impedance. It could be interesting to test this setup with a range of masters and slaves to see if there is actually a correlation between the difference in impedance of the devices and the stability problems encountered in this work upon contact.

8.5.5 Transparency

When it comes to transparency, it is evident that there is a trade off between transparency and stability. When scaleing the force down, the robot does not bump back to the same degree or even at all, depending on

the stiffness of the material to which it is in contact. What is more preferable to prioritize when it comes to an ultrasound setup? If the system is going to work on patients, the obvious answer is of course stability. We do not want a system that suddenly "attacks" our patient. In the previous subsection I asked whether it is necessary to have full transparency of the force in such a setup. If one of the goals of this setup is to help radiologists avoid carpatel tunnel syndrom and other injuries due to extensive use of high force it is not necessarialy desireable to replicate the force at it's full. It might be preferable to have a downscaled force as an aid to help them know when they are in contact with the patient and to a certain amount how much. In order to have some control of how much force is actually exerted to the patient, a color bar displaying the true force could be implemented. This should then be placed somewhere the radiologists looks, either in the same image as the ultrasound, or perhaps at the robot iself; perhaps both so that it is visual for the radiologist when he/she has the attention at both the patient and the screen.

The speed is also subject to scaleing. In this setup, i have proposed a down-scaling of the speed when in touch with the environment in order to increase stability and to get better accuracy when doing the ultrasound diagnostics. I can see different scenarios in a diagnostisation process where different settings would be prereable. In a transportingfase, getting from a to b, it would be desirable to do that as fast as possible, it might even be okay to scale the speed up. Then when starting the ultrasound session it would be preferable to scale the speed down. Not only would this cope with the contact instability previous mentioned, but it would probably help increase the accuracy of the ultrasound imaging. In a finished setup i imagine the operator sitting next to a wheel or slider where it is possible to adjust this scale in real-time as he/she feels appropriate.

8.5.6 Workspace

Regarding the workspace, there is an obvious difference in scale. It is not a problem per se, with the implementation of relative position and orientation. However with a haptic device with a larger workspace it would not be necessary to "reset" the position as much as it has to be done today. Today we click a button in order to teleoperate, and the teleoperation stops once we release the button. This way we can reuse the Phantom Omni's workspace many times. This however breaks the workflow a bit, especially if the operator is in the middle of an ultra-sound session. In the long run, this solution is not optimal and a larger workspace would be prefered. Another solution would be to scale the velocity up. However scaleing things up also means scaleing up noise present in the system which might lead to undesired behavior.

8.5.7 Choice of common coordinate system

The slave base system was chosen as the common coordinate system in this implementation. As the phantom was situated in the XY-plane of this

coordinate system, the same as our master, operating this was very intuitive. However in the real world this might not be the case. Patients are not flat objects, and especially not a pregnant stomach. A better solution in these situations could be to have the common coordinate system to be the tool-frame of the slave. This way, the orientation of the tool would be the biased orientation. Having this orientation relative to a surface in a random direction might be easier and more intuitive. However this means that left and right translational movement in the master will be different directions in the master based on the toolframe orientation. This could be an interesting topic to further investigate to see which of the approaches is more intuitive for different situations.

8.5.8 Noise

Noise filtering had to be implemented in order to get the force feedback to a level where it was useful. The came from the force sensor, where small vibrations in the robot manifested themselves in the force sensor as vibrations. Some times, it seemed as if the slave was not able to get to it's set point but was vibrating back and forth around a joint angle for a certian joint. A small push helped it to get in place. A viable cause for this phenomena is that the robot was stuck very close to an encoder state change, so close that is was below the minimal angle the joint can rotate. These high-frequent noise signals made it impossible to keep the master still. The noise was even noticable when the robot was moving as the table/chart is currently mounted on would react to the inertia of the robot while it was moving. A better suitable mounting would probably help to get rid of some of this noise.

8.5.9 Time delay

Although the effects of time delay have not been tested in this setup, a small discussion on the matter will take place.

Actually performed telesurgery

When dealing with teleoperation in a lab environment, delay is rarely an issue these days. However if you extend the applications out of the lab you might encounter setups that will suffer from delay, which gives an extra challenge regarding both transparency and stability. When [10] did their transatlantic telesurgery experiment they had 2 dedicated fiberoptic channels between New York and Strasbourg. This minimized the communication latency, but is a very costly approach. This was the first teleosurgery on a human subject, and thus a milestone within the field of telesurgery. Another interesting thing about the setup in this surgery is the fact that they used the Zeus operating robot as it was. They did not do any modification with the robot in order to make it more suitable for long distance telesurgery. Instead they built a communication layer around it, meaning the algorithms involved in coping with the communication delay

were independent of the robot at hand. This was all part of the research, as it would be more cost-effective to use a readily available setup instead of modifying one. This is true both for aquisition, service and upgrading; it makes the system more flexible. In Canada, dr M. Anvari has performed a number of telesurgeries on human patients [3] with great success. An already existing network connection existing between hospital networks in Canada was used as communication infrastructure. This shows that it is possible to use telesurgery in a clinical setup with existing communication infrastructure.

Time delay on teleoperation

In a teleoperation setup, there are many things that can contribute to delay. The communication channel and signal processing at both ends are two obvious reasons. The UDP protocol is the commonly used protocol for transferring information between the master and the slave. Today there two ways of sending data over a network that are most commonly in use. The transport control protocol (TCP) and the user datagram protocol (UDP). The transport control protocol is a confirmation-based protocol, which guarantees the data to arrive at it's destination by resending the data until reception is confirmed. User datagram protocol on the other hand does not guarantee this, resulting in diminished overhead and less delay as no retransmitting of packages is involved. [33] showed this in an experiment where a sine wave was sent with both protocols from Atlanta, Georgia to Metz, France. Another challenge regarding packet delay is asynchronity. If sending a packet in a lab network this is probably not noticeable, but if the packets are to go over the internet packets can take many ways to get to the target. [28] proposes a Network Delay Regulator to cope with this. This is essentially achieved by implementing a packet queue at the receiving ends of the communication layer which distributes the packets evenly to the master/slave. Doing so makes sure that noe packets writes over it's predecessor if they arrive almost simultanously.

To a certain degree, humans are capable of adapting to delay in teleoperation. However it is shown that delays past 300 milliseconds makes it almost impossible to operate [55]. There are different approaches beeing looked into to cope with time delay. Implementing a predicting interface is one, Here it is attempted to predict the next system state, or states, in order to "go around" the time delay [2] [11]. Another approach is modeling the signals as wave variables [33].

It is not just the velocity and force that needs to be transmitted in such a setup. Just as important is the visual feedback, which needs a lot more bandwith [30]. However a few things can be tuned regarding the video in order to devcrease the delay. The framerate can be turned down, as well as the bitrate. The video encoding can also be chosen specifically to minimize delay.

Motivation for my setup

Having to cope with time delay comes from the desire to perform teleoperation over large distances. For the ultrasound robot i have been working on, there is a clear motivation to this kind of setup out to the districts. As ultrasound diagnostics get more and more specialized it is hard to keep personell with necessary comptence throughout the districts. It can be seen today how hospitals and communal doctor centers are beeing merged, leading to longer way to go for the patients. With the ultrasound robot it is imaginably to install this at doctors offices throughout the districts, and then having a team of specialized operators teleoperating them from a clinic elsewhere. This could be imagined beeing done for more diagnstic methods than ultrasound, but ultrasound is a good place to start considering the none-invasiveness of it's nature. In norway the internet infrastructure is very well built, so it should be fairly easy to get the necessary bandwith for this kind of setup.

Part III

Conclusion

Chapter 9

Conclusion

9.1 Conclusion

A bilateral master-slave system has been implemented with haptic feedback and implemented to a robot system for medical purposes. It has been showed that teleoperating a UR5 robot equipped with a force-torque sensor and an ultrasound probe benefits from having haptic feedback. The haptic feedback helps the user know how much force is exerted on the environment, and to keep the ultrasound probe more stable when contact is established with the patient. Allthough the data in the experiments were not statistically significant, they give a good indication in the mentioned direction. Haptic feedback should be implemented in the finished setup. It has also been pointed out the benefits of such a setup both with respect to the radiologists doing ultrasound examinations today, as well as the benefits of long-distance ultrasound examination. This setup was designed and implemented using commercially abailable hardware. This has proved to work fine, thus using such equipment is a viable option to designing an entire robot system from scratch.

9.2 Future work

More research into this setup is required in order to and up with a setup possible to use on patients. A better suited master controller is needed. The Phantom Omni is to small and to weak to be useful in such a setup. With the new haptic device incorporated into the system, more user tests should be performed, preferably with a more diversified background as in this test. A force strength indicator should be developed so that it is possible to have a visual control of how much force is actually exerted to the patient at runtime. More reearch should also be conducted to find better ways to decide whether or not the slave is in contact with the patient. External sensors, like a kinect, could be incorporated in order to better the decision making. More research could also be done to see if stability would improve by implementing the 3 or 4 channel lawrence architecture instead of the 2 channel when it comes to the moment of contact.

Bibliography

[1] *Datasheet of Sensable Phantom Omni.*

[2] D Andreau, P Fraisse, and J.A. Segovia De Los Rios. Teleoperation over an IP Network: from control to architectural considerations. In *2004 8th International Conference on Control, Automation, Robotics and Vision, Kunmin, China*, pages 765–770. IEEE, 2004.

[3] M Anvari. Remote telepresence surgery: the canadian experience. *Surgical Endoscopy*, 21:537–541, 2007.

[4] E. Aoki, T. Suzuki, E. Kobayashi, I. Sakuma, K. Kozo, and M. Hashizume. Modular design of master-slave surgical robotic system with reliable real-time control performance. In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 80 –86, feb. 2006.

[5] J. Arata, H. Takahashi, P. Pitakwatchara, S. Warisawa, K. Konishi, K. Tanoue, S. Ieiri, S. Shimizu, N. Nakashima, K. Okamura, Young Soo Kim, Sung Min Kim, Joon-Soo Hahm, M. Hashizume, and M. Mitsuishi. A remote surgery experiment between japan-korea using the minimally invasive surgical system. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 257–262, 2006.

[6] P. Arbeille, J. Ruiz, P. Herve, M. Chevillot, G. Poisson, and F. Perrotin. Fetal tele-echography using a robotic arm and a satellite link. *Ultrasound in Obstetrics and Gynecology*, 26(3):221–226, 2005.

[7] ATI Industrial Automation. Ati gamma force / torque product site. July 2013. http://www.ati-ia.com/products/ft/ft_models.aspx?id=Gamma.

[8] Sarmad Aziz. Development and verification of ground-based tele-robotics operations concept for dextre. *Acta Astronautica*, 86(0):1 – 9, 2013.

[9] L.A. Ballard, S. Sabanovic, J. Kaur, and S. Milojevic. George charles devol, jr. [history]. *Robotics Automation Magazine, IEEE*, 19(3):114 –119, sept. 2012.

[10] S.E. Butner and M. Ghodoussi. Transforming a surgical robot for human telesurgery. *Robotics and Automation, IEEE Transactions on*, 19(5):818–824, 2003.

[11] Alessandro Casavola and Michaela Sorbara. Towards constrained teleoperation for safe long-distance robotic surgical operations. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain*, pages 685–690, 2005.

[12] Chae G. An Christopher G. Atkeson and John M. Hollerbach. Estimation of inertial parameters of manipulator loads and links. *The International Journal of Robotic Research*, 5(3):101–119, 1986.

[13] R. Clarke. Asimov's laws of robotics: implications for information technology-part i. *Computer*, 26(12):53 –61, dec 1993.

[14] M. Cote, J.-A. Boulay, B. Ozell, H. Labelle, and C.-E. Aubin. Virtual reality simulator for scoliosis surgery training: Transatlantic collaborative tests. In *Haptic Audio visual Environments and Games, 2008. HAVE 2008. IEEE International Workshop on*, pages 1 –6, oct. 2008.

[15] F. Courreges, P. Vieyres, and R. S H Istepanian. Advances in robotic tele-echography services - the otelo system. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, volume 2, pages 5371–5374, 2004.

[16] Force Dimension. Force dimension's website. *Force Dimension's website*, 2012. "http://www.forcedimension.com".

[17] A.G. Gallagher, N. McClure, J. McGuian, K. Ritchie, and Sheehy N.P. An ergonomic analysis of the fulcrum effect in the aquisition of endoscopic skills. *Endoscopy*, 30(7):617–620, 1998.

[18] Geomagic. Geomagic acquires sensable 3d design and haptics business. *Press release*, apr 2012.

[19] A. Gourdon, P. Poignet, G. Poisson, Pierre Vieyres, and P. Marche. A new robotic mechanism for medical application. In *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*, pages 33–38, 1999.

[20] Gwang Min Gu, Don Ju Lee, and Jung Kim. Development of a low cost force sensor for wearable robotic systems. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 1450 –1455, dec. 2011.

[21] Butterfly Haptics. Butterfly haptics' website. *Butterfly Haptics' website*, 2012. "http://www.butterflyhaptics.com/".

[22] Haption. Haption's website. *Haption's website*, 2012. "http://www.haption.com/site/index.php/en/".

[23] Peter F. Hokayem and Mark W. Spong. Bilateral teleoperation: An historical survey. *Automatica*, 42:2035 – 2057, apr 2006.

[24] Brian P. MD Jacob and Michael MD FACS FRCSC Gagner. Robotics and general surgery. *Surgical Clinics of North America*, 83(6):1405–1419, 2003.

[25] N. Koizumi, S. Warisawa, M. Nagoshi, H. Hashizume, and M. Mitsuishi. Construction methodology for a remote ultrasound diagnostic system. *Robotics, IEEE Transactions on*, 25(3):522–538, 2009.

[26] D. Kubus and F M Wahl. Scaling and eliminating non-contact forces and torques to improve bilateral teleoperation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ Internation Conference on*, pages 5133 –5139, oct 2009.

[27] Martin Kvalbein. The use of a 3d sensor (kinect) for robot motion compensation, the applicability in relation to medical applications. Master's thesis, University of Oslo, Norway, 2012.

[28] A Leleve, P Fraisse, and P Dauchez. Telerobotics over IP Networks: Towards a Low-level Real-time Architecture. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii*, pages 643–648. IEEE, 2001.

[29] MD PhD Liselotte Mettler. From air insufflation to robotic endoscopic surgery: A rocky road. *Journal of Minimally Invasive Gynecology*, 18(3):275 – 283, may 2011.

[30] Mitchell J.H Lum, Jacob Rosen, Hawkeye King, Diana C.W Friedman, Lendvay Thomas S., Andrew S. Wright, Mika N. Sinanan, and Blake Hannaford. Teleoperation in Surgical Robotics - Network Latency Effects on Surgical Performance. In *31st Annual International Conference of the IEEE EMBS, Minneapolis, Minnesota*, pages 6860–6863. IEEE, 2009.

[31] K. Masuda, E. Kimura, N. Tateishi, and K. Ishihara. Three dimensional motion mechanism of ultrasound probe and its application for tele-echography system. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 2, pages 1112–1116 vol.2, 2001.

[32] T. Meiss, C. Budelmann, T.A. Kern, S. Sindlinger, C. Minamisava, and R. Werthschutzky. Intravascular palpation and haptic feedback during angioplasty. In *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pages 380 –381, march 2009.

[33] Saghir Mumir and J. Wayne Book. Internet-based teleoperation using wave variables with prediction. *IEEE/ASME Transaction on mechatronics*, 7(2):124–133, 2002.

[34] Declan G Murphy, Raymond Tong, Rajiv Goel, and Anthony J. Costello. Robotic technology in surgery: Current status in 2008. *ANZ Journal of Surgery*, 78(12):1076–1081, 2008.

[35] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC press, electronic edition, 1994.

[36] E. Naerum, J. Cornella, and O.J. Elle. Contact force estimation for backdrivable robotic manipulators with coupled friction. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3021 –3027, sept. 2008.

[37] E. Naerum, J. Cornella, and O.J. Elle. Wavelet networks for estimation of coupled friction in robotic manipulators. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 862 –867, may 2008.

[38] E. Naerum and B. Hannaford. Global transparency analysis of the lawrence teleoperator architecture. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 4344 –4349, may 2009.

[39] Edvard Naerum, Ole Jakob Elle, and Blake Hannaford. The effect of interaction force estimation on performance in bilateral teleoperation. *Haptics, IEEE Transactions on*, 5(2):160 –171, april-june 2012.

[40] Nasa. Image of dextre robot on international space station. July 2013. http://spaceflight.nasa.gov/gallery/images/station/crew-27/hires/iss027e016182.jpg.

[41] Kiyoshi Ohishi, Masaru Miyazaki, and Masahiro Fujita. Hybrid control of force and position without force sensor. In *Industrial Electronics, Control, Automation, 1992. Power Electronics and Motion Control, Proceedings of the 1992 International Congress on*, volume 2, pages 670–675, 1992.

[42] François Pierrot, Etienne Dombre, Eric Dégoulange, Loïc Urbain, Pierre Caron, Jérome Gariépy, and Jean louis Mégnien. Hippocrate: A safe robot arm for medical applications with force feedback. *Medical Image Analysis*, 3:285–300, 1999.

[43] Oxford University Press. Oxford english dictionary online. *Oxford English Dictionary Online*, Apr 2012. http://www.oed.com/view/Entry/242646?redirectedFrom=teleoperator#eid.

[44] Oxford University Press. Oxford english dictionary online. *Oxford English Dictionary Online*, Mar 2012. http://www.oed.com/view/Entry/84082?redirectedFrom=haptic#eid.

[45] A.M. Priester, S. Natarajan, and M.O. Culjat. Robotic ultrasound systems in medicine. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 60(3):–, 2013.

[46] P. Puangmali, K. Althoefer, L.D. Seneviratne, D. Murphy, and P. Dasgupta. State-of-the-art in force and tactile sensing for minimally invasive surgery. *Sensors Journal, IEEE*, 8(4):371 –381, april 2008.

[47] R. Rayman, S. Primak, R. Patel, M. Moallem, R. Morady, M. Tavakoli, V. Subotic, N. Galbraith, A. van Wynsberghe, and K. Croome. Effects of latency on telesurgery: an experimental study. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2005*, pages 57–64, 2005.

[48] F. Rehnmark, I. Spain, W. Bluethmann, M. Goza, R.O. Ambrose, and K. Alder. An experimental investigation of robotic spacewalking. In *Humanoid Robots, 2004 4th IEEE/RAS International Conference on*, volume 1, pages 366 – 384 Vol. 1, nov. 2004.

[49] Universal Robotics. Image of universal robot. July 2013. http://media1.limitless.dk/UR_Robot/UR5_Robot_with_controller.jpg.

[50] Jee-Hwan Ryu, Dong-Soo Kwon, and B. Hannaford. Stable teleoperation with time-domain passivity control. *Robotics and Automation, IEEE Transactions on*, 20(2):365–373, 2004.

[51] S. Salcudean, G. Bell, S. Bachmann, W. Zhu, P. Abolmaesumi, and P. Lawrence. Robot-assisted diagnostic ultrasound–design and feasibility experiments. In *Medical Image Computing and Computer-Assisted Intervention-MICCAI 99*, pages 1062–1071. Springer, 1999.

[52] M. Sato. Spidar and virtual reality. In *Automation Congress, 2002 Proceedings of the 5th Biannual World*, volume 13, pages 17 – 23, 2002.

[53] U. Seibold, B. Kubler, and G. Hirzinger. Prototype of instrument for minimally invasive surgery with 6-axis force sensing capability. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 496 – 501, april 2005.

[54] Sensable. Sensable's website. *Sensable's website*, 2012. "http://www.sensable.com/".

[55] Thomas B. Sheridan. Space teleoperation through time delay: Review and prognosis. *IEEE Transaction on Robotics and Automation*, 9(5):592–606, 1993.

[56] A.C. Smith and K. Hashtrudi-Zaad. Application of neural networks in inverse dynamics based contact force estimation. In *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, pages 1021 –1026, aug. 2005.

[57] Gyung Tak Sung and Inderbir S. Gill. Robotic laparoscopic surgery: A comparison of the da vinci and zeus systems. *Urology*, 58(6):893–898, 2001.

[58] Intuitive Surgical. Intuitive surgical annual report 2010. *Intuitive Surgical's website*, 2010. "http://investor.intuitivesurgical.com/phoenix.zhtml?c=122359&p=irol-reportsAnnual.

[59] University of Alberta TBS-group, July 2013. http://www.ece.ualberta.ca/~tbs/Full-blockdiag-orig.png.

[60] Sensable Technologies. Image of sensable technologies' phantom omni. July 2013. http://www.dentsable.com/documents/images/LargePHANTOMOmniImage.jpg.

[61] Dapeng Tian and K. Ohnishi. Online identification and compensation of the force estimating error in sensor-less bilateral control system. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1 –4, sept. 2010.

[62] Linda Jacoba Martina van den Bedem. *Realization of a Demonstrator Slave for Robotic Minimally Invasive Surgery*. PhD thesis, Technische Universiteit Eindhoven, 2010.

[63] H.E. Vanderpool, E.A. Friis, B.S. Smith, and K.L Harms. Prevalence of carpal tunnel synsdrome and other work-related musculoskeletal problems in cardiac sonographers. *Journal of Occupational Medicine*, 35:604–610, 1993.

[64] A. Vilchis, J. Troccaz, P. Cinquin, K. Masuda, and F. Pellissier. A new robot architecture for tele-echography. *Robotics and Automation, IEEE Transactions on*, 19(5):922–926, 2003.

[65] Vilchis-Gonzalez, J.C. Avila-Vilchis, and A. Garcia-Torres. Termi robot. In *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007*, pages 464–469, 2007.

[66] Declan Walsh. Us extends drone strikes to somalia. *The Guardian Online*, jun 2011. http://www.guardian.co.uk/world/2011/jun/30/us-drone-strikes-somalia.

[67] Qiong Wang, Hui Chen, Wen Wu, Hai Yang Jin, and Pheng Ann Heng. A virtual surgical simulator for mandibular angle reduction based on patient specific data. In *Virtual Reality Workshops (VR), 2012 IEEE*, pages 85 –86, march 2012.

[68] Alexei Wedmid, Elton Llukani, and David I. Lee. Future perspectives in robotic surgery. *BJU International*, 108(6b):1028–1036, 2011. http://dx.doi.org/10.1111/j.1464-410X.2011.10458.x.

Part IV

Appendix

Chapter A

Haptic device comparison table

Producer
Product name
Degrees of freedom
Force Feedback
Maximum volume
Translation workspace
Rotation workspace
Maximum force
Continous force
Maximum torque
Continous torque
Backdrive friction
Maximum stiffness translation
Maximum stiffness rotation
Minimum resolution translation
Miminum resolution rotation
Interface

Table A.1: *Values of rows in device tables*

Haption	Haption	Sensable	Sensable
Virtuose 6D desktop	Virtuose 6D	The PHANTOM Desktop Device	The PHANTOM Omni Device
6	6	6	6
6 DOF	6 DOF	3 DOF (x,y,z)	3 DOF (x,y,z)
300x260 mm	900x600x1060 mm	160x120x120 mm	160x120x70 mm
Sphere with radius 60 mm	Corresponds to movement of human arm	Hand movement pivoting at wrist	Hand movement pivoting at wrist
35 degrees in the 3 directions at the centre	Same as above	Same as above	Same as above
5 to 15 N	35 N	7.9 N	3.3 N
1.4 - 3 N	10 N	1.75 N	0.88 N
0.2 - 0.5 Nm	3 Nm	na	na
0.06 to 0.14 Nm	1 Nm	na	na
na	na	0.06 N	0.26 N
2500 N/m	2000 N/m	1.86N/mm, y - 2.35N/mm, z - 1.48N/mm	x - 1.26N/mm y - 2.31 N/mm, z - 1.02 N/mm
2 Nm / rad	30 Nm / rad	na	na
0.1 mm	na	0.023 mm	0.055 mm
0.05 degrees	na	na	na
na	na	Parallel port, firewire	Firewire

Table A.2: *Specifications for Haption and Sensable devices*

Sensable	Sensable	Sensable	Sensable
Premium 1.0	Premium 1.5/6DOF	Premium 1.5 High Force / 6DOF	Premium 3.0/6DOF
6	6	6	6
3 DOF (x,y,z)	3 DOF (x,y,z)	6 DOF	6 DOF
254x178x127 mm	381x267x191 mm	381x267x191 mm	838x584x406 mm
Hand movement pivoting at wrist	Lower arm movement, pivoting at elbow	Lower arm movement pivoting at elbow	Full arm movement pivoting at shoulder
Same as above	Yaw: 297d, Pitch: 260d, Roll: 335d	Yaw: 297d, Pitch: 260d, Roll: 335d	Same as above
8.5 N	8.5 N	37.5 N	22 N
1.4 N	1.4 N	6.2 N	3 N
na	Yaw/Pitch: 515 mNm, Roll: 170 mNm	Yaw/Pitch: 515mNm, Roll 170 mNm	Yaw/Pitch: 515 mNm, Roll: 170 mNm
na	Yaw/Pitch: 515 mNm Roll: 170 mNm	Yaw/Pitch: 188 mNm Roll: 48 mNm	Yaw/Pitch: 188 mNm Roll: 48 mNm
0.04 N	0.04N	0.2 N	0.2 N
3.5 N/mm	3.5 N/mm	3.5 N/mm	1 N/mm
na	na	na	na
0.03 mm	0.03 mm	0.007 mm	0.02 mm
Yaw/Pitch: 0.0023d Roll: 0.008d	Yaw/Pitch: 0.0023d Roll: 0.008d	Yaw/Pitch: 0.0023d Roll: 0.008d	Yaw/Pitch: 0.0023d 0.0023d, Roll: 0.008d
Parallel Port	Parallel port	Parallel Port	Parallel Port

Table A.3: *Specifications for Sensable devices*

Force Dimension	Force Dimension	Force Dimension	Force Dimension	Force Dimension	Force Dimension
Delta 3	Delta 6	Omega 3	Omega 6	Omega 7	Sigma 7
3	6	3	6	7	7
3 DOF (x,y,z)	6 DOF	3 DOF (x,y,z)	3 DOF (x,y,z)	7 DOF	7 DOF
400x260 mm	400x260 mm	160x110 mm	160x110 mm	160x110 mm	190x130 mm
Sphere	Sphere	Sphere	Sphere	Sphere, grasping 25 mm	Sphere, grasping 25 mm
na	na	na	240x140x320 degrees	240x140x180 degrees	235x140x200 degrees
20 N	20 N	12 N	12 N	12 N, grasping 8 N	20 N, grasping 8 N
na	na	na	na	na	na
na	150 mNm	na	na	na	400 mNm
na	na	na	na	na	na
na	na	na	14.5 N/mm	14.5 N/mm	14.5 N/mm
na	na	na	na	na	na
< 0.01 mm	< 0.01 mm	< 0.01 mm	< 0.01 mm	< 0.01 mm, grasping 0.006	< 0.012 mm, grasping 0.006 mm
na	< 0.04 degrees	na	0.09 degrees	0.09 degrees	0.013 degrees
USB 2.0	USB 2.0	USB 2.0	USB 2.0	USB 2.0	USB 2.0

Table A.4: *Specifications for Force Dimension devices*

Chapter B

System response

B.1 Bode plots

B.1.1 Transparent

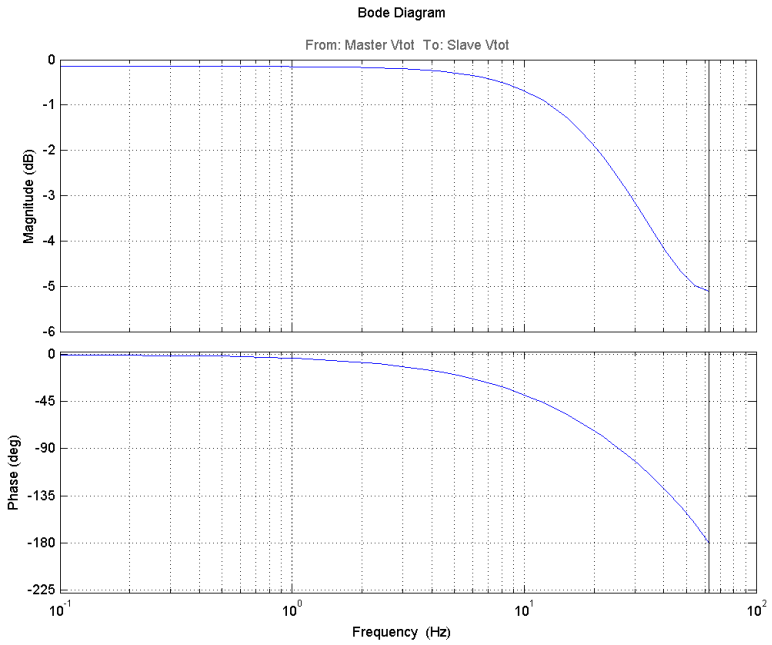


Figure B.1: *Bode plot for absolute velocity throughput in transparent controller in free space.*

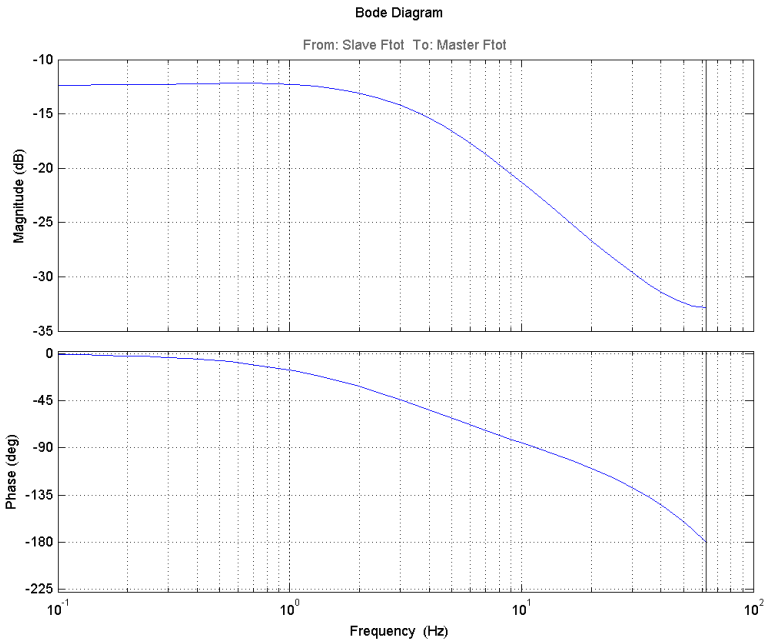


Figure B.2: *Bode plot for absolute force throughput in transparent controller touching phantom.*

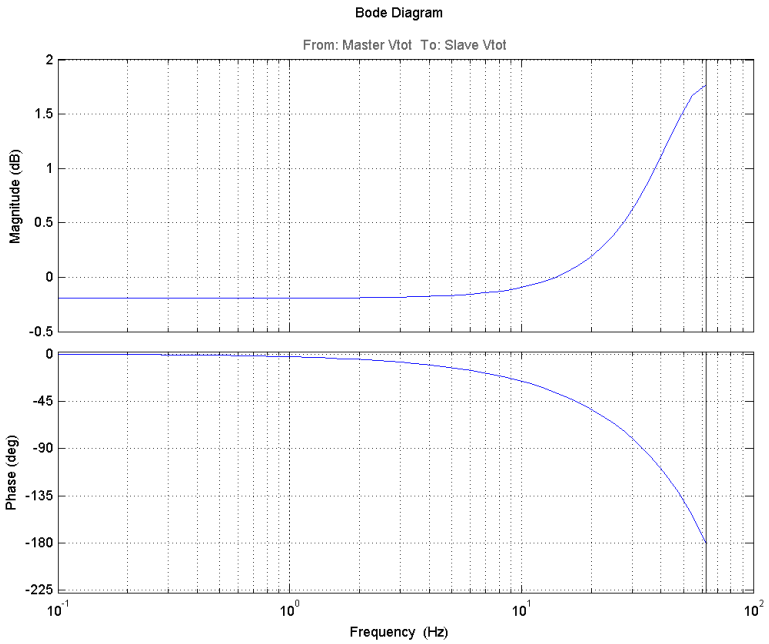


Figure B.3: *Bode plot for absolute velocity throughput in transparent controller touching phantom.*

B.1.2 Static low

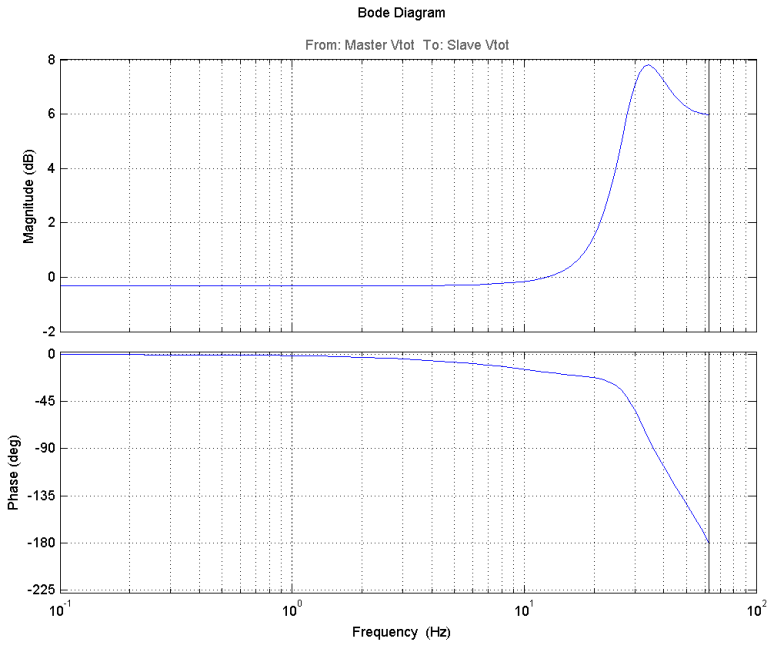


Figure B.4: *Bode plot for absolute velocity throughput in static low controller in free space.*

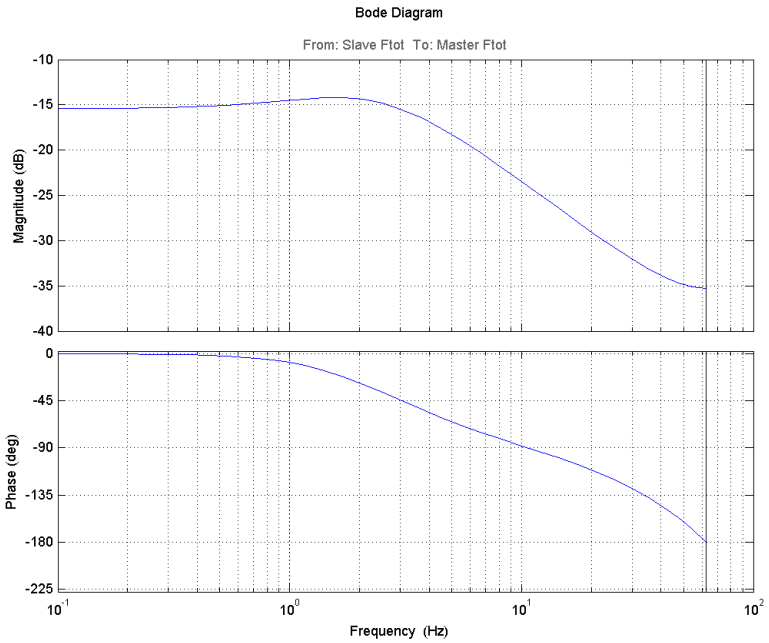


Figure B.5: *Bode plot for absolute force throughput in static low controller touching phantom.*

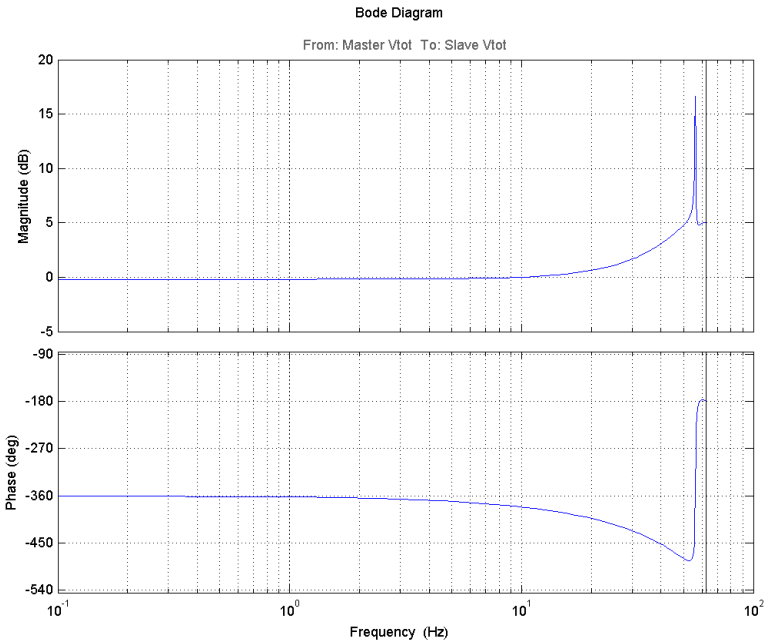


Figure B.6: *Bode plot for absolute velocity throughput in static low controller touching phantom.*

B.1.3 Sigmoid Zero

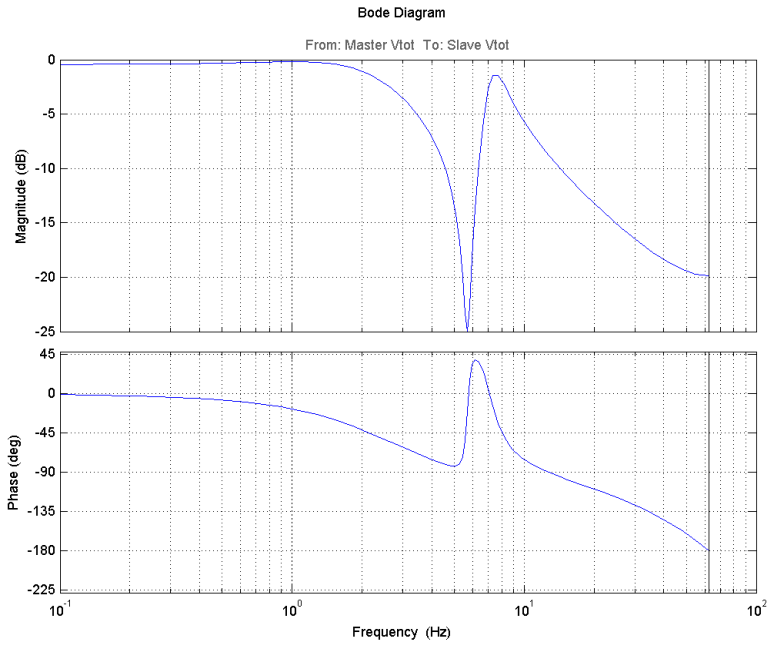


Figure B.7: *Bode plot for absolute velocity throughput in sigmoid zero controller in free space.*

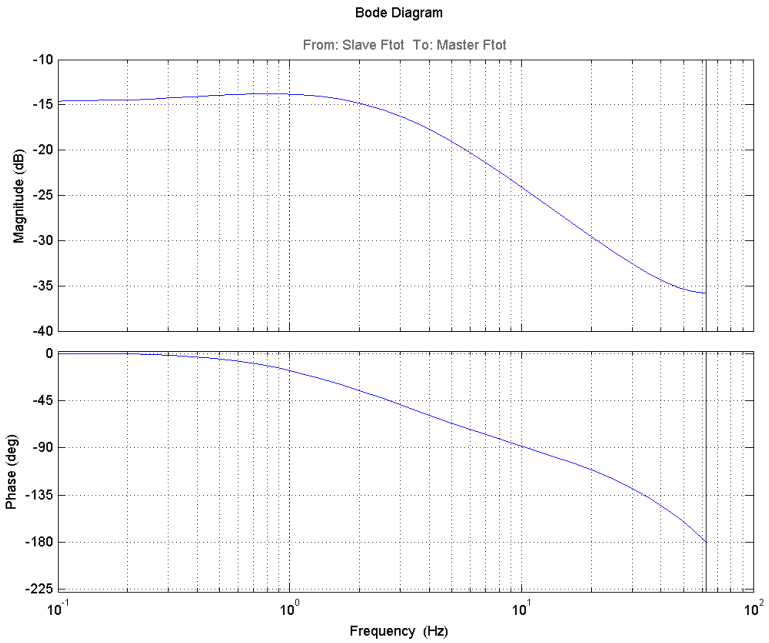


Figure B.8: *Bode plot for absolute force throughput in sigmoid zero controller touching phantom.*

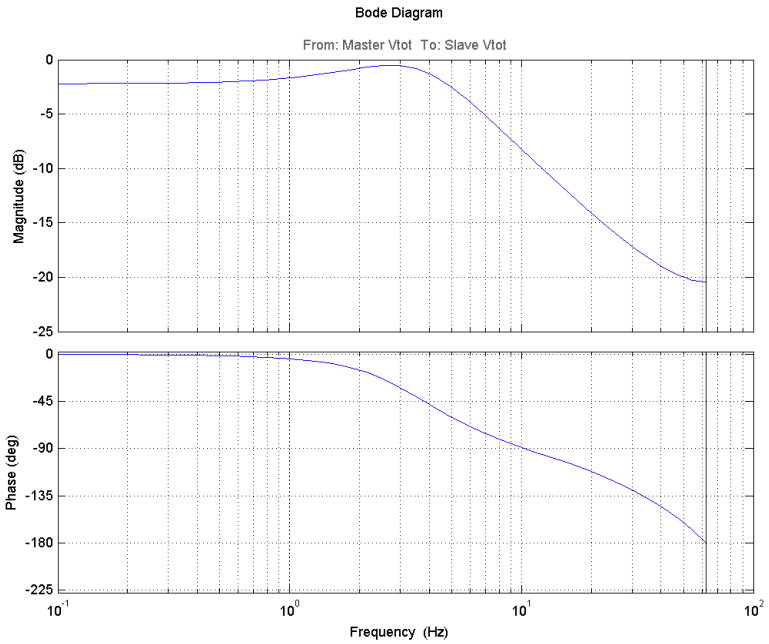


Figure B.9: *Bode plot for absolute velocity throughput in sigmoid zero controller touching phantom.*

B.1.4 Sigmoid Low

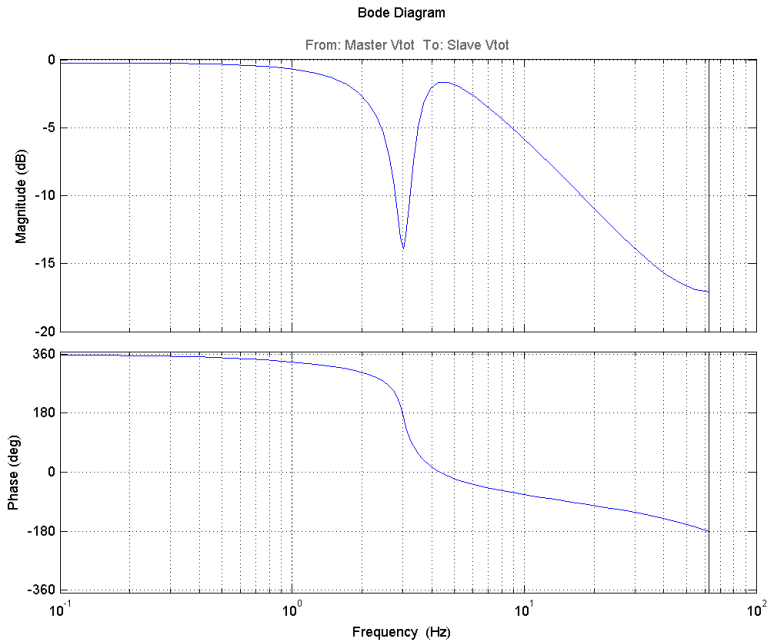


Figure B.10: *Bode plot for absolute velocity throughput in sigmoid low controller in free space.*

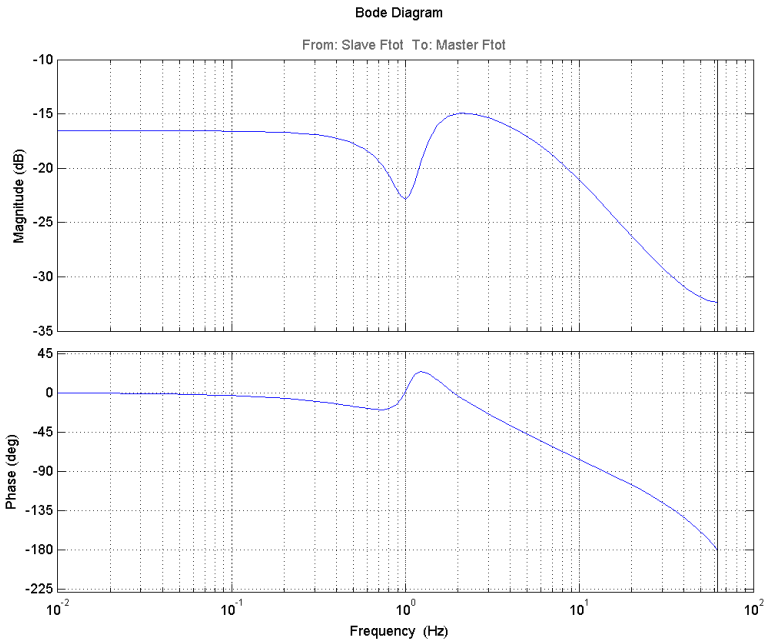


Figure B.11: *Bode plot for absolute force throughput in sigmoid low controller touching phantom.*

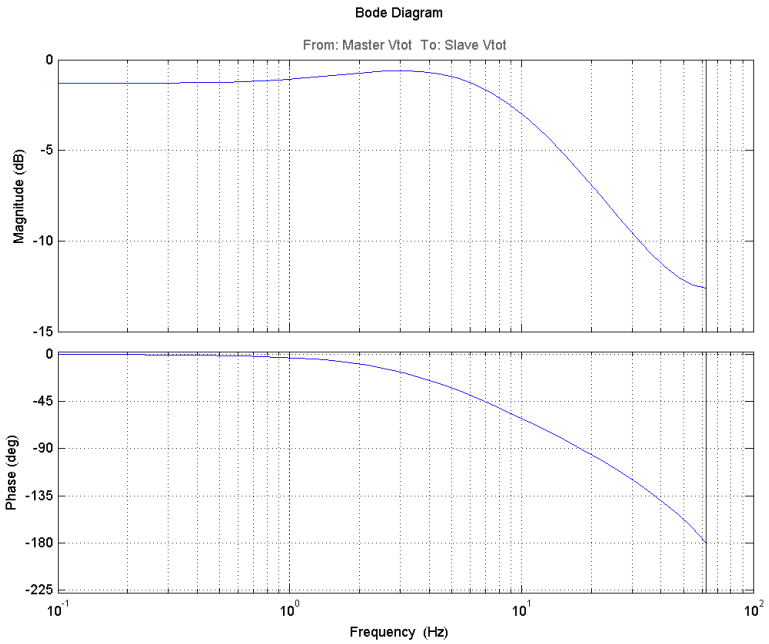


Figure B.12: *Bode plot for absolute velocity throughput in sigmoid low controller touching phantom.*

B.1.5 Without haptic

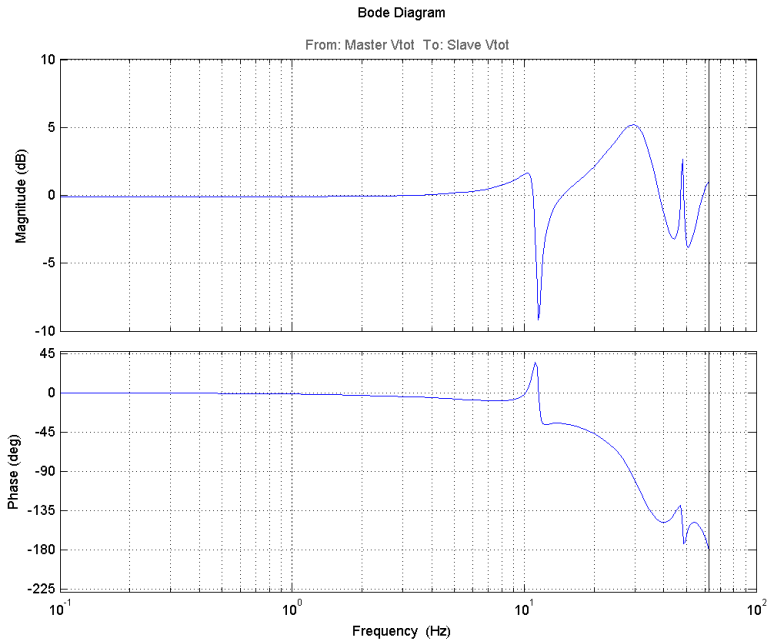


Figure B.13: *Bode plot for absolute velocity throughput in no haptic controller moving in free space.*

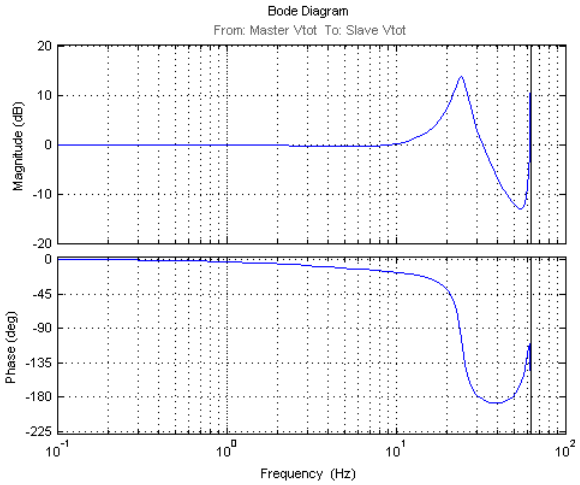


Figure B.14: *Bode plot for absolute velocity throughput in no haptics controller touching phantom.*

B.1.6 Angular velocity - no haptic

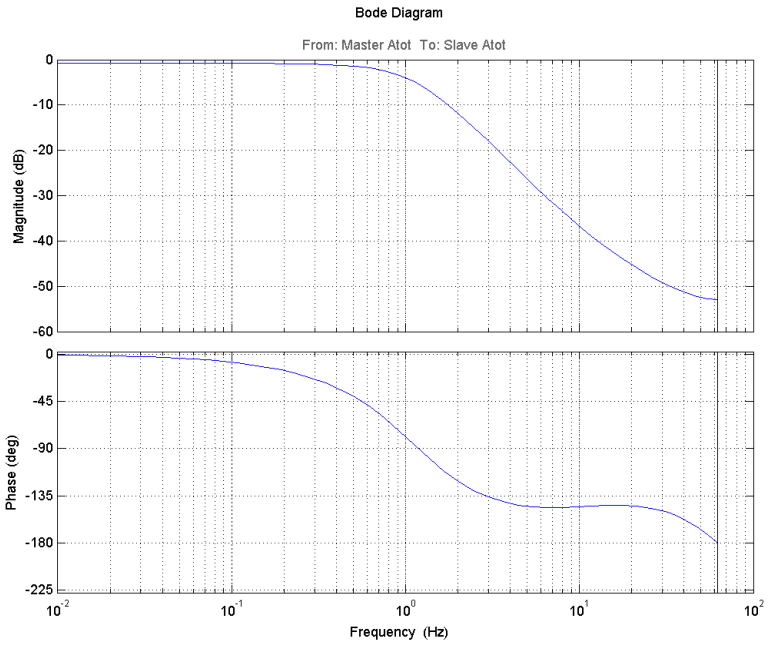


Figure B.15: *Bode plot for angular velocity throughput in no haptic controller in free space total angular velocity.*

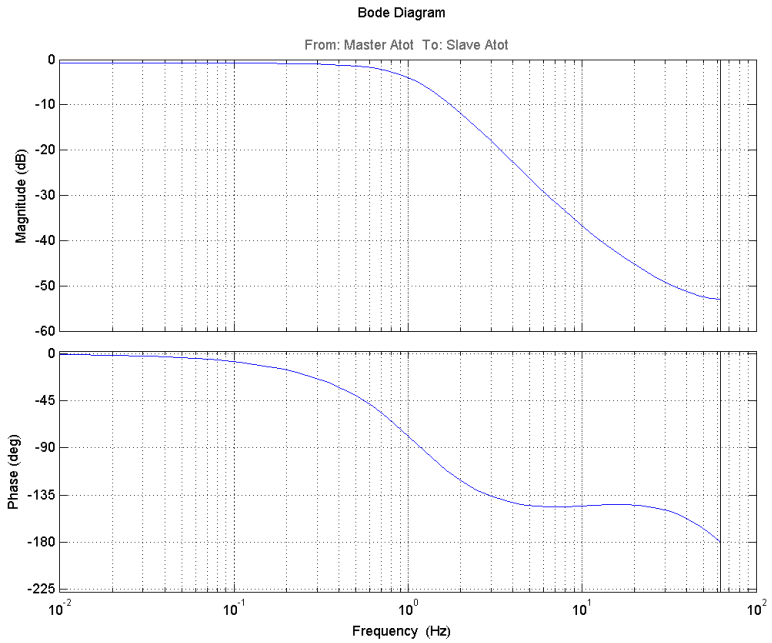


Figure B.16: *Bode plot for angular velocity throughput in no haptic controller touching phantom total angular velocity.*

B.1.7 Angular velocity - haptic feedback

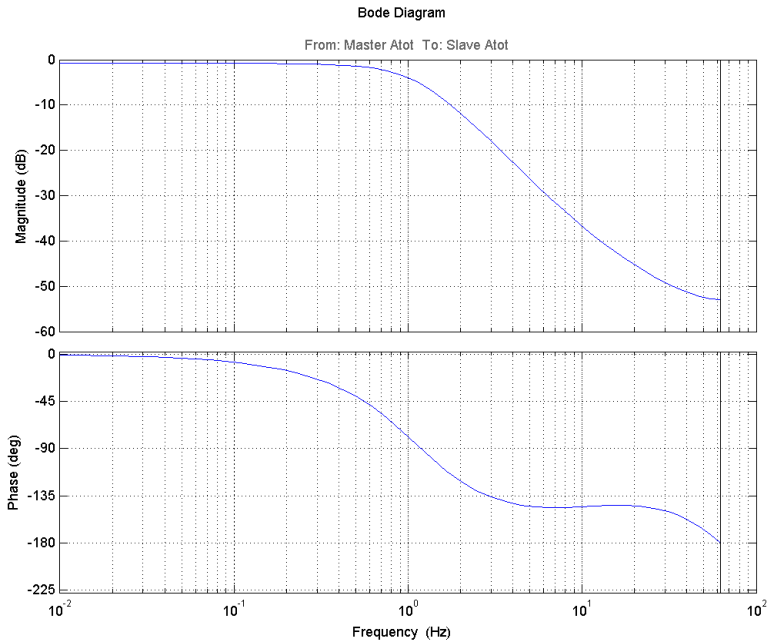


Figure B.17: *Bode plot for angular velocity throughput in transparent controller in free space total angular velocity.*

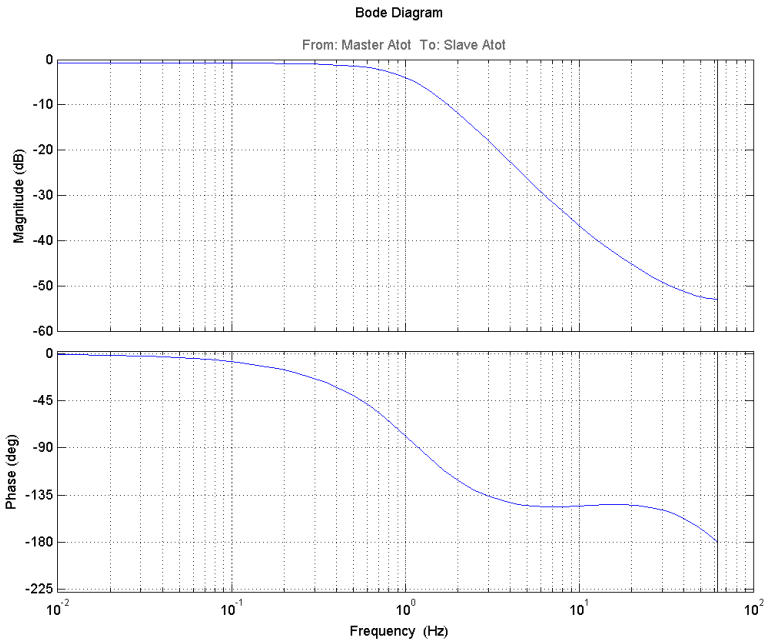


Figure B.18: *Bode plot for angular velocity throughput in transparent controller touching phantom total angular velocity.*

Chapter C

Questionnaire

Figure 1: Sphere within phantom.

Evaluation form of teleoperation with haptic feedback

This is an evaluation form for the experiment you are about to do. First you will perform a regular ultrasound-examination. Then you will do the same thing with a teleoperation setup, without Haptic feedback. Then at the end we will do the same thing with haptic feedback.
First some information about you:

About you

1. **Your name**: _____
2. **How old are you?** I am _____ years old.
3. **What is your profession?** _____
4. **Define your experience performing ultrasound diagnostics:** First time □—□—□—□—□—□ Expert

The Task

In this experiment, your task is to locate a certain item within the phantom using ultrasound. The item is a sphere, and can be recognized as in the image in figure 1. You will try this first with your own hands, then with the help of the teleoperator.

Conventional Ultrasound

5. **How easy was it to get to the target:**
   Not easy □—□—□—□—□ Easy
6. **How easy was it to anticipate the exerted force on the phantom:**
   Not easy □—□—□—□—□ Easy

Teleoperation without haptic feedback

7. **How was the overall feeling of the system:**
   Unstable □—□—□—□—□ Stable
8. **Was the response to your movement:**
   Uncontrollable □—□—□—□—□ Good
9. **How easy was it to get to the target:**
   Not easy □—□—□—□—□ Easy
10. **How easy was it to anticipate the exerted force on the phantom:**
    Not easy □—□—□—□—□ Easy
11. **Did the robot do what you wanted it to?**
    No □—□ Yes
12. **If no, please elaborate:**

Teleoperation with haptic feedback - Transparent

13. **How was the overall feeling of the system:**
    Unstable □—□—□—□—□ Stable
14. **Was the response to your movement:**
    Uncontrollable □—□—□—□—□ Good
15. **How easy was it to get to the target:**
    Not easy □—□—□—□—□ Easy
16. **How easy was it to anticipate the exerted force on the phantom:**
    Not easy □—□—□—□—□ Easy
17. **Did the robot do what you wanted it to?**
    No □—□ Yes
18. **If no, please elaborate:**

Teleoperation with haptic feedback - Low

19. **How was the overall feeling of the system:**
    Unstable □—□—□—□—□ Stable
20. **Was the response to your movement:**
    Uncontrollable □—□—□—□—□ Good
21. **How easy was it to get to the target:**
    Not easy □—□—□—□—□ Easy
22. **How easy was it to anticipate the exerted force on the phantom:**
    Not easy □—□—□—□—□ Easy
23. **Did the robot do what you wanted it to?**
    No □—□ Yes
24. **If no, please elaborate:**

Teleoperation with haptic feedback - Sigmoid zero

25. **How was the overall feeling of the system:**
    Unstable □—□—□—□—□ Stable
26. **Was the response to your movement:**
    Uncontrollable □—□—□—□—□ Good
27. **How easy was it to get to the target:**
    Not easy □—□—□—□—□ Easy
28. **How easy was it to anticipate the exerted force on the phantom:**
    Not easy □—□—□—□—□ Easy
29. **Did the robot do what you wanted it to?**
    No □—□ Yes
30. **If no, please elaborate:**

Teleoperation with haptic feedback - Sigmoid low

31. **How was the overall feeling of the system:**
    Unstable □—□—□—□—□ Stable
32. **Was the response to your movement:**
    Uncontrollable □—□—□—□—□ Good
33. **How easy was it to get to the target:**
    Not easy □—□—□—□—□ Easy
34. **How easy was it to anticipate the exerted force on the phantom:**
    Not easy □—□—□—□—□ Easy
35. **Did the robot do what you wanted it to?**
    No □—□ Yes
36. **If no, please elaborate:**

Objective observations

Conventional Ultrasound

37. Time spent: _____

38. Accuracy: Clumsy □—□—□—□—□ Spot on

Teleoperation without haptic feedback - Transparent

39. Time spent: _____

40. Accuracy: Clumsy □—□—□—□—□ Spot on

Teleoperation with haptic feedback - Transparent

41. Time spent: _____

42. Accuracy: Clumsy □—□—□—□—□ Spot on

Teleoperation with haptic feedback - Low

43. Time spent: _____

44. Accuracy: Clumsy □—□—□—□—□ Spot on

Teleoperation with haptic feedback - Sigmoid zero

45. Time spent: _____

46. Accuracy: Clumsy □—□—□—□—□ Spot on

Teleoperation with haptic feedback - Sigmoid low

47. Time spent: _____

48. Accuracy: Clumsy □—□—□—□—□ Spot on

Chapter D

Questionnaire results

D.1 Info on subjects

	Age	Occupation	US experience
Subject 1	28	PhD student, robotics	3
Subject 2	32	Researcher, robotics	4
Subject 3	27	Researcher, image analysis	1
Subject 4	34	Researcher	2
Subject 5	25	Researcher	1
Myself	27	Masterstudent	3

Table D.1: *Subjects performing the tests*

D.2 Time spent

[H]

	S1	S2	S3	S4	S5	Me	Avg
Conventional ultrasound	0:15	0:35	0:40	0:43	0:37	2:01	0:33,6
Transparent	3:45	4:13	1:11	0:57	5:01	1:29	3:01,4
Static Low	1:08	0:16	3:28	2:47	2:03	1:01	1:56,4
Sigmoid Zero	0:16	0:32	0:42	0:41	0:36	0:36	0:33,4
Sigmoid low	1:17	5:01	2:17	0:17	1:03	0:37	1:59,0
Without haptic feedback	0:37	1:19	0:47	3:12	0:51	1:17	1:21,2

Table D.2: *Subjects' time spent*

D.3 Accuracy

	S1	S2	S3	S4	S5	Me	Avg
Conventional ultrasound	5	4	3	4	4	4	4,0
Transparent	2	1	4	4	2	5	2,6
Static Low	3	4	4	4	4	4	3,8
Sigmoid Zero	5	3	5	3	5	5	4,2
Sigmoid low	4	2	4	5	4	5	3,8
Without haptic feedback	4	3	4	2	4	4	3,4

Table D.3: *Accuracy pr subject*

D.4 Overall feeling of the system

	S1	S2	S3	S4	S5	Me	Avg
Conventional ultrasound	na	na	na	na	na	na	na
Transparent	3	5	4	4	2	4	3,6
Static Low	4	5	3	3	4	4	3,8
Sigmoid Zero	4	5	4	4	5	4	4,4
Sigmoid low	4	4	4	4	4	5	4,0
Without haptic feedback	5	5	4	5	5	5	4,8

Table D.4: *Overall feeling of system per subject*

D.5 Response to movement

	S1	S2	S3	S4	S5	Me	Avg
Conventional ultrasound	na	na	na	na	na	na	na
Transparent	3	3	5	4	2	5	3,4
Static Low	4	5	3	3	4	4	3,8
Sigmoid Zero	4	3	5	2	4	5	3,6
Sigmoid low	4	3	4	4	4	5	3,8
Without haptic feedback	5	5	5	5	5	5	5,0

Table D.5: *Response to movement per subject*

D.6 Finding the target

	S1	S2	S3	S4	S5	Me	Avg
Conventional ultrasound	4	5	4	4	4	3	4,2
Transparent	2	3	4	4	1	4	2,8
Static Low	3	5	3	2	3	4	3,2
Sigmoid Zero	3	3	5	3	5	4	3,8
Sigmoid low	3	3	4	4	4	4	3,6
Without haptic feedback	4	5	5	2	5	5	4,2

Table D.6: *Evaluation of difficulty finding the target per subject*

D.7 Anticipation of exerted force

	S1	S2	S3	S4	S5	Me	Avg
Conventional ultrasound	5	5	3	na	3	4	4,0
Transparent	2	4	3	4	3	4	3,2
Static Low	3	5	4	2	4	4	3,6
Sigmoid Zero	3	3	5	3	4	5	3,6
Sigmoid low	3	3	5	4	4	5	3,8
Without haptic feedback	4	1	2	1	5	3	2,6

Table D.7: *Anticipation of exerted force per user*

D.8 Robot behaviour

	S1	S2	S3	S4	S5	Myself
Conventional ultrasound	na	na	na	na	na	na
Transparent	No	Yes	Yes	Yes	No	Yes
Static Low	Yes/no	Yes	No	Yes	No	Yes
Sigmoid Zero	Yes/no	Yes	Yes	Yes	Yes	Yes
Sigmoid low	Yes/no	Yes	Yes	Yes	Yes	Yes
Without haptic feedback	Yes	Yes	Yes	Yes	Yes	Yes

Table D.8: *Evaluation of robot behaviour per subject*

D.9 Elaboration

	S1	S2	S3	S4	S5	Myself
Conventional ultrasound	na	na	na	na	na	na
Transparent	1.)	na	na	6.)	10.)	na
Static Low	2.)	na	5.)	7.)	11.)	na
Sigmoid Zero	3.)	na	na	8.)	na	na
Sigmoid low	4.)	na	na	na	na	na
Without haptic feedback	na	na	na	9.)	na	na

Table D.9: *Specification of subject evaluation*

	Controller	Elaboration
1.)	Transparent	Hard to keep the contact and get the probe orthogonal to the phantom.
2.)	Static Low	A bit the same as number 1.
3.)	Sigmoid Zero	The same as 2.
4.)	Sigmoid Low	The same as 2.
5.)	Static Low	At the ends it was stopping abruptly.
6.)	Transparent	Yaw a bit difficult.
7.)	Static Low	Bumping up and down.
8.)	Sigmoid Zero	Difficult to control angles.
9.)	Without haptic feedback	Hard not to crush the patient.
10.)	Transparent	It is like the coordinate system changed.
11.)	Static Low	Better than before, but still hard to control.

Table D.10: *Subjects evaluation in words*

Chapter E

Code

E.1 jef-haptics.cpp

```
#include <stdio.h>
#include <assert.h>

#include <string.h>

#include "jef-haptics.h"

#define LOGGFIL "JEF_logg.txt"
#define HAPTICFILE "Haptic_time.dat"
#define ROBOTFILE "Robot_time.dat"

HHD hhd;

void initialize_time_logg(){

        FILE *filr, *filh;
        filr = fopen(HAPTICFILE,"w+");
        filh = fopen(ROBOTFILE,"w+");
        fclose(filr);
        fclose(filh);
}

void robot_time_logg(double timestamp){
```

```
        FILE *fil;
        fil = fopen(ROBOTFILE,"a+");
        fprintf(fil,"%.8f\n", timestamp);
        fclose(fil);
}

void haptic_time_logg(double timestamp){



        FILE *fil;
        fil = fopen(HAPTICFILE,"a+");
        fprintf(fil,"%.8f\n", timestamp);
        fclose(fil);
}


void initialize_logg(){

        FILE *fil;
        fil = fopen(LOGGFIL,"w+");
        fclose(fil);

}
```

```
void logg(      double rfx, double rfy, double rfz,
                double fx, double fy, double fz,
                double my_vx, double my_vy, double my_vz,
                double vx, double vy, double vz,
                double Posx, double Posy, double Posz,
                double my_fx, double my_fy, double my_fz,
                double p1x, double p1y, double p1z,
                double p2x, double p2y, double p2z,
                double EinMx, double EinMy, double EinMz,
                double EoutMx, double EoutMy, double EoutMz,
                double EinSx, double EinSy, double EinSz,
                double EoutSx, double EoutSy, double EoutSz
        ) {

        FILE *fil;
        fil = fopen(LOGGFIL,"a+");
        fprintf(fil,"%.8f %.8f %.8f %.8f %.8f %.8f ", rfx, rfy, rfz, fx, fy, fz);
        fprintf(fil,"%.8f %.8f %.8f %.8f %.8f %.8f ", my_vx, my_vy, my_vz, my_fx, my_fy, my_fz);
        fprintf(fil,"%.8f %.8f %.8f %.8f %.8f %.8f ", p1x, p1y, p1z, p2x, p2y, p2z);
        fprintf(fil,"%.8f %.8f %.8f %.8f %.8f %.8f ", vx, vy, vz, Posx, Posy, Posz);
        fprintf(fil,"%.8f %.8f %.8f %.8f %.8f %.8f ", EinMx, EinMy, EinMz, EoutMx, EoutMy, EoutMz);
        fprintf(fil,"%.8f %.8f %.8f %.8f %.8f %.8f \n", EinSx, EinSy, EinSz, EoutSx, EoutSy, EoutSz);
        fclose(fil);

}
```

```
int CalculateOmega(mat33 dR, vec3 *out){


        mat33 omega, ones_3, eye_3;
        cx_vec lambda;
        cx_mat V;

        ones_3.ones();
        eye_3.eye();

        eig_gen(lambda,V,dR);

        omega = real(V*log(ones_3-eye_3+diagmat(lambda))*inv(V))/(1./125);

//      cout << "submarker ...\n";
        out[0](0) = (omega(2,1)-omega(1,2))/2.;
        out[0](1) = (omega(0,2)-omega(2,0))/2.;
        out[0](2) = (omega(1,0)-omega(0,1))/2.;

        return 0;

}
```

```
int InitializeHapticDevice(){

        HDErrorInfo error;

        hhd = hdInitDevice(HD_DEFAULT_DEVICE);
        if (HD_DEVICE_ERROR(error = hdGetError())){

                hduPrintError(stderr, &error, "Failed to initialize haptic device.");
                fprintf(stderr, "\n Exiting.\n");

                return -1;
        }

        hdEnable(HD_FORCE_OUTPUT);
                hdStartScheduler();

        if (HD_DEVICE_ERROR(error = hdGetError())){

                hduPrintError(stderr, &error, "Failed to start scheduler\n");
                fprintf(stderr, "Exiting.\n");
                return -1;
        }

        printf("Found device model: %s\nNow initialized.\n\n", hdGetString(HD_DEVICE_MODEL_TYPE));

        return 0;
```

```
}

int ForceMean(struct HapticPack *hp){

        double temp[3] = {0,0,0};
        int j, i;

        for(j=0; j< 3; j++){
                for(i=0; i< (WINDOW-1); i++){

                        hp->f_mean[j][i] = hp->f_mean[j][i+1];
                        temp[j] = temp[j] + hp->f_mean[j][i]/WINDOW;
                }
                hp->f_mean[j][(WINDOW-1)] = hp->f[j];
                temp[j] = temp[j] + hp->f_mean[j][(WINDOW-1)]/WINDOW;

        }

        hp->f[0] = temp[0];
        hp->f[1] = temp[1];
        hp->f[2] = temp[2];

        return 0;
}
```

```
int StopHapticDevice(){

        hdStopScheduler();

        return 0;

}

int HapticCall(struct HapticPack *hp){


        double buffer;

        int i,j;

        hdScheduleSynchronous(JEFHapticLink, hp, HD_DEFAULT_SCHEDULER_PRIORITY);
        HapticJEFfiltering(hp);
        buffer = hp->hdf[0];
        hp->hdf[0] = hp->hdf[1];
        hp->hdf[1] = hp->hdf[2];
        hp->hdf[2] = buffer;

        HapticJEFController(hp, TRANSPARENT);

        return 0;
```

```
}

double JEF_low_pass(double x,double xm, double xmm, double xmmm){

        double f = xm + ALPHA*(x-xm);
        double result = (f + xm + xmm) / 3;

        return result;

}

int HapticJEFController(struct HapticPack *hp, int controller){


        double d;
        double d2;

        double low;
        double totf = sqrt(pow(hp->hdf[1],2) + pow(hp->hdf[1],2) + pow(hp->hdf[2],2));
        double kf, kv, k;

        int flim = 6;

        k = 1;
        kf = 1;
        kv = 1;
```

```
for(int i = 0; i < 3; i++){

        switch(controller){

                case TRANSPARENT:

                        break;

                case STATIC_LOW:

                        low = 0.4;

                        if(abs(hp->hdf[i]) > JEF_LIMIT){
                                kv = low;
                        }

                        break;

                case SIGMOID_ZERO:

                        if((hp->hdf[i]*hp->hdv[i]) < 0){
//                      if(abs(hp->hdf[i]) > 2){

                                k = 1./(1+exp(abs(hp->hdf[i])-4));
```

```
                kv = k;

            }

            break;

        case SIGMOID_LOW:

            low = 1 - 0.4;

            if((hp->hdf[i]*hp->hdv[i]) < 0){

                k = 1./(1+exp(abs(hp->hdf[i])-4));
                kv = k*low + 0.4;

            }

            break;

        case NO_HAPTIC_FEEDBACK:

            kf = 0;

        default:
            break;
```

```
                }

                hp->hdf1[i] = hp->hdf[i]*kf*FORCESCALE;
                hp->v[i] = hp->hdv[i]*kv*VELOCITYSCALE;

//              cout << "kv: "<< kv << " " << hp->hdf[i] << " " << hp->hdf1[i] << " " << hp->v[i] << "\n";

        }

}

int HapticJEFfiltering(struct HapticPack *hp){

        int i,j;
        double force;
        double filtforce[3];

        if(hp->bt){

                for(i=0;i<12;i++ ){hp->haptic_bias[i] = hp->T[i];}
                for(i=12;i<15;i++ ){hp->haptic_bias[i] = 0;}
                for(i=15;i<16;i++ ){hp->haptic_bias[i] = 1;}
        }

                for (i=0; i<3; i++){
```

```
                force = hp->f[0]*hp->transf(0,i) + hp->f[1]*hp->transf(1,i) + hp->f[2]*hp->transf(2,i);
                filtforce[i] = JEF_low_pass(force,hp->hdlf[i],hp->hdllf[i],hp->hdlllf[i]);
            }

        hp->hdf[0] = filtforce[0];
        hp->hdf[1] = filtforce[1];
        hp->hdf[2] = filtforce[2];

        return 0;
}

HDCallbackCode HDCALLBACK JEFHapticLink(void *data){

        HDErrorInfo error;
        HDint butt;

        double vel2[3],vel[3];
        double cvel[6];
        double angular[3], jvel[3], gvel[3];
        double thisp[3], lastp[3];
        double tg[3], lg[3];
        double tj[3], lj[3];
        double force[3];

        int i;
```

```
double temp[16],T[16],Tl[16];


struct HapticPack *hp = (struct HapticPack *) data;

hdBeginFrame(hhd);

hdGetIntegerv(HD_CURRENT_BUTTONS,&butt);

double fmax = 3.3;
double fscale = 1;

if(butt != 2){for(i=0;i<3;i++){force[i] = 0;}}
else {

        for(i=0;i<3;i++){
                force[i] = hp->hdf1[i];
        }

}

hdGetDoublev(HD_CURRENT_TRANSFORM,T);
hdGetDoublev(HD_LAST_TRANSFORM,Tl);

hdGetDoublev(HD_CURRENT_POSITION,thisp);
hdGetDoublev(HD_LAST_POSITION,lastp);
```

```
hdSetDoublev(HD_CURRENT_FORCE, force);
hdGetDoublev(HD_CURRENT_GIMBAL_ANGLES, tg);
hdGetDoublev(HD_LAST_GIMBAL_ANGLES, lg);
hdGetDoublev(HD_CURRENT_JOINT_ANGLES, tj);
hdGetDoublev(HD_LAST_JOINT_ANGLES, lj);

hp->pos[0] = thisp[0];
hp->pos[1] = thisp[1];
hp->pos[2] = thisp[2];

if (butt != 2){
        hp->bt = 1;
        vel = {0, 0, 0};
        angular = {0, 0, 0};
        cvel = {0, 0, 0, 0, 0, 0};
}else{
        hp->bt = 0;
        for(i=0;i<3;i++){
                vel[i] = T[12+i] - Tl[12+i];
                T[12+i] = 0;


                vel2[i] = thisp[i] - lastp[i];
                jvel[i] = tj[i] - lj[i];
                gvel[i] = tg[i] - lg[i];
        }
```

```
}

hdEndFrame(hhd);

for(i=0; i<16; i++){hp->T[i] =  T[i];}

hp->hT  << T[0] << T[4] <<  T[8] << 0 << endr
        << T[1] << T[5] <<  T[9] << 0 << endr
        << T[2] << T[6] << T[10] << 0 << endr
        << T[3] << T[7] << T[11] << 1 << endr;


hp->hdlllv[0] = hp->hdllv[0];
hp->hdlllv[1] = hp->hdllv[1];
hp->hdlllv[2] = hp->hdllv[2];
hp->hdllv[0] = hp->hdlv[0];
hp->hdllv[1] = hp->hdlv[1];
hp->hdllv[2] = hp->hdlv[2];
hp->hdlv[0] = hp->hdv[0];
hp->hdlv[1] = hp->hdv[1];
hp->hdlv[2] = hp->hdv[2];
hp->hdv[0] = JEF_low_pass(vel[2],hp->hdlv[0], hp->hdllv[0], hp->hdlllv[0]);
hp->hdv[1] = JEF_low_pass(vel[0],hp->hdlv[1], hp->hdllv[1], hp->hdlllv[1]);
hp->hdv[2] = JEF_low_pass(vel[1],hp->hdlv[2], hp->hdllv[2], hp->hdlllv[2]);
```

```
        if (HD_DEVICE_ERROR(error = hdGetError())){
            hduPrintError(stderr, &error, "Error detected while reading velocity.\n");
        }

        return HD_CALLBACK_DONE;

    }
```

E.2 jef-haptics.h

```
#include <stdio.h>
#include <assert.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <sys/time.h>
#include <sys/mman.h>
#include <signal.h>

#include <native/syscall.h>
#include <native/task.h>
#include <native/types.h>
#include <native/timer.h>
#include <native/mutex.h>
#include <native/pipe.h>
#include <native/queue.h>

#include <armadillo>

#include "ur-linux-tweaks.h"
#include "ur-control.h"

#include <HD/hd.h>
#include <HDU/hduError.h>
```

```
#include <HDU/hduVector.h>
#include <HDU/hduGenericMatrix.h>
#include <HDU/hduMatrix.h>
#include <HDU/hduMath.h>

#define VELOCITYSCALE 1./100
//#define VELOCITYSCALE 1.5
#define ANGULARVELOCITYSCALE 1
#define FORCESCALE 1000./1000
#define HAPTICPERIODE 50000 // 1000000

#define WINDOW 1

#define CUTOFF 55
#define SAMPLE_RATE 500 // 1000
#define RC 1.0/(CUTOFF*2*3.14)
#define DT 1.0/(SAMPLE_RATE)
#define ALPHA DT/(RC+DT)

#define JEF_LIMIT 3

#define TRANSPARENT 1
#define STATIC_LOW 2
#define SIGMOID_ZERO 4
#define SIGMOID_LOW 5
#define NO_HAPTIC_FEEDBACK 8
```

```
using namespace arma;
using namespace hduGenericMatrix;

struct HapticPack
{
        RT_MUTEX v_lock;
        RT_MUTEX f_lock;


        RT_MUTEX hdv_lock;
        RT_MUTEX hdlv_lock;
        RT_MUTEX hdf_lock;

        RT_MUTEX T_lock;
        RT_MUTEX transf_lock;


        double f_mean[3][WINDOW];

        vec6 v;
        vec6 ur_v;
        vec6 vsum;
        vec6 f;
        vec6 fl;
```

```
        hduVector3Dd hdv;
        hduVector3Dd hdlv;
        hduVector3Dd hdllv;
        hduVector3Dd hdlllv;
        hduVector3Dd hdf;
        hduVector3Dd hdlf;
        hduVector3Dd hdllf;
        hduVector3Dd hdlllf;

        hduVector3Dd hdf1;
        hduVector3Dd hdlf1;
        hduVector3Dd hdf2;

//      hduVector3Dd E;
//      hduVector3Dd El;

        double EinM[3];
        double EinS[3];
        double EoutM[3];
        double EoutS[3];

        double my_f[3];
        double my_v[3];
        double p1[3];
        double p2[3];
```

```
        hduVector3Dd pos;
        hduVector3Dd wall;

        vec3 sang;
        vec3 mang;


        mat33 sc;
        mat33 sl;

        mat33 mc;
        mat33 ml;

        double T[16];
        mat44 transf;
        mat44 hT;
        mat44 ur_T;
        mat44 robot_bias;
        mat44 haptic_bias;

        int bt;
};

HDCallbackCode HDCALLBACK JEFHapticLink(void *data);
```

```
int InitializeHapticDevice();
int StopHapticDevice();
int HapticCall(struct HapticPack hp);
int HapticCall(struct HapticPack *hp);
int CalculateOmega(mat33 dR, vec3 *out);
int HapticJEFfiltering(struct HapticPack *hp);
int HapticJEFController(struct HapticPack *hp, int controller);

int Passivity_Based_Controller(struct HapticPack *hp);
double PBC_alpha(double w, double v);
int ForceMean(struct HapticPack *hp);


void initialize_logg();
void initialize_time_logg();
void haptic_time_logg(double timestamp);
void robot_time_logg(double timestamp);
void logg(      double rfx, double rfy, double rfz, double fx, double fy, double fz,
                double my_vx, double my_vy, double my_vz, double vx, double vy, double vz,
                double Posx, double Posy, double Posz, double my_fx, double my_fy, double my_fz,
                double p1x, double p1y, double p1z, double p2x, double p2y, double p2z,
                double EinMx, double EinMy, double EinMz, double EoutMx, double EoutMy, double EoutMz,
                double EinSx, double EinSy, double EinSz, double EoutSx, double EoutSy, double EoutSz
          );
```

E.3 ur-jef-controller.cpp

```
#include <stdio.h>
#include <stdio.h>
#include <assert.h>
#include <ctime>
#include <time.h>
#include <iostream>

#include <string.h>

#include "ur-kinematics.h"
#include "jef-haptics.h"

int running;

struct ur_robot* robot;
struct shared_variables* share;
struct HapticPack hp;
vec6 hapticgain;

RT_MUTEX hp_lock;


RT_TASK rt_external_robot_controller;
```

```
RT_TASK rt_external_haptic_controller;

void rt_task_external_haptic_controller(void* cookie){

        int error;
        unsigned long overrun = 0;

        double last,current,max;

        max = 0;

        rt_task_set_periodic(NULL,TM_NOW,HAPTICPERIODE);

        while(running){

                rt_mutex_acquire(&hp_lock,TM_INFINITE);

                timespec ts;
                ts.tv_sec = 0;
                ts.tv_nsec = 0;
                clock_gettime(CLOCK_REALTIME, &ts);
//              clock_t start = clock();
                last = ts.tv_nsec;

                HapticCall(&hp);
```

```
                clock_gettime(CLOCK_REALTIME,&ts);
                current = ts.tv_nsec;


//              printf("Haptics loop: %0.4f ms\n", (((double)clock()-start) / CLOCKS_PER_SEC)*1000*1000);
//              cout << "Haptics loop: " << (((double)clock()-start) / CLOCKS_PER_SEC)*1000*1000 << "\n";
//              cout << "Haptics loop: " << ((double)clock()-start) << "\n";


                cout << "Time␣taken␣is:␣" << ts.tv_sec << "␣" << ts.tv_nsec << "\n";
                cout << "Time␣taken␣is:␣" << current << "␣-␣" << last << "␣=␣" <<  \\
                                (current - last) << "Max:␣" << max << "\n";

                cout << "Time␣taken␣is:␣" << (((current-last)/CLOCKS_PER_SEC)*1000*1000) \\
                         << "-" << CLOCKS_PER_SEC << "\n";

                rt_mutex_release(&hp_lock);

                haptic_time_logg(((current-last)/CLOCKS_PER_SEC)*1000*1000);

                if(max < (current-last)){max = current - last;}

                error = rt_task_wait_period(&overrun);

        }
```

```
}

void rt_task_external_robot_controller(void* cookie){

        int rt_error = ur_linux_cpuaffinity_all(8);
        printf("CPU addinity returned %i\n", rt_error);

        int myC = 0;

        int error;
        RT_QUEUE ctrl_queue;

        printf("Opening rt queue %s\n", UR_EXT_CTRL_QUEUE);
        error = rt_queue_bind(&ctrl_queue, UR_EXT_CTRL_QUEUE, TM_INFINITE);
        if (error < 0){

                printf("%s(%i): rt_queue_bind had error: %s\n",__FUNCTION__,__LINE__,strerror(-error));
                rt_task_delete(NULL);
        } else {
                printf("RT control queue bind successfully (%i)\n", error);
        }


        int i;
```

```
for(i=0; i<3; i++){hp.hdf[i] = 0; hp.f[i] = 0; hp.f[i+3] = 0;}

float ctrl = 0;

mat44 M_6_F, M_F_T, O_T_R, M_6_T_INV, SPEIL;

O_T_R   << 0 << 1 << 0 << 0 << endr
        << 0 << 0 << 1 << 0 << endr
        << 1 << 0 << 0 << 0 << endr
        << 0 << 0 << 0 << 1 << endr;

SPEIL   << -1 <<  0 << 0 << 0 << endr
        <<  0 << -1 << 0 << 0 << endr
        <<  0 <<  0 << 1 << 0 << endr
        <<  0 <<  0 << 0 << 1 << endr;


  double t_dg = 27.89; // Should be positive?
  double t = (t_dg/360)*3.14;
  M_6_F.eye();
  M_6_F(2,3) = 0.00436;
  M_F_T.eye();
  /* Us probe */

  M_F_T << 1.0 << 0.0    << 0.0     << 0.0      << endr
        << 0.0 << cos(t) << -sin(t) << 0.0282   << endr   // 0.0282
```

```
        << 0.0 << sin(t) << cos(t)  << 0.1715   << endr  // 0.1715
        << 0.0 << 0.0    << 0.0     << 1      << endr;

  setTransformationMatrix_6_F(M_6_F);
  setTransformationMatrix_F_T(M_F_T);

M_6_T_INV = inv(M_6_F*M_F_T);

double Fm[6] = {0, 0, 0, 0, 0, 0};

double current, last, max;
max = 0;

while (running) {


        timespec ts;
        clock_gettime(CLOCK_REALTIME,&ts);
        last = ts.tv_nsec;

        struct ur_msg_info_ext data;

        ssize_t size = rt_queue_read(&ctrl_queue, &data, sizeof(data), TM_INFINITE);

        if ( size == sizeof(data) ){
```

```
mat66 J_bb;
mat66 J6_bb;
mat66 T_ft;
mat44 ur_T, my_urT, my_Trans, my_Trans_Inv;
mat44 ur_6T, v_M;
mat44 ur_1T, ur_2T;
mat44 ur_3T, ur_4T;
mat44 ur_5T;
mat44 set_T;
mat44 B, Xh, Xr;
mat33 set_C, set_L;
vec6 qd, q_meas, vqd, v_c, f_filt;
struct ur_msg_cmd_ext cmd;
struct ur_storage_tracker stor;
struct ur_msg_cmd test;

double f_sum;


rt_mutex_acquire(&hp.f_lock,TM_INFINITE);
vec6 F(data.ft.force_torque);

my_Trans <<  0 << -1 <<  0 << 0 << endr
         <<  1 <<  0 <<  0 << 0 << endr
         <<  0 <<  0 << -1 << 0 << endr
```

```
            <<  0 <<  0 <<  0 << 1 << endr;

my_Trans_Inv = inv(my_Trans);

hp.fl = hp.f;
hp.f = F;

hp.f[0] = -hp.f[0];
hp.f[1] = -hp.f[1];

ForceMean(&hp);


hp.hdlllf = hp.hdllf;
hp.hdllf = hp.hdlf;
hp.hdlf = hp.hdf;
hp.hdlf1 = hp.hdf1;
//          hp.El = hp.E;
f_filt[0] = hp.hdf1[2];
f_filt[1] = hp.hdf1[0];
f_filt[2] = -hp.hdf1[1];

rt_mutex_release(&hp.f_lock);

int j;
```

```
for(i = 0; i < 3; i++){
        for(j=0;j<3;j++){
                hp.sl(i,j)=  hp.ur_T(i,j);
        }
}


getForwardKinematrics_0_F(data.status.q, &ur_T);

if(hp.bt){
        hp.ur_T = ur_T;
}

array2vector(data.status.q, &q_meas,6);

if(hp.bt){
        hp.robot_bias = hp.ur_T;
        hp.robot_bias[12] = 0;
        hp.robot_bias[13] = 0;
        hp.robot_bias[14] = 0;
}

my_urT = hp.ur_T;
my_urT[12] = 0;
my_urT[13] = 0;
my_urT[14] = 0;
```

```
mat33 dRm;
mat33 dRs;
mat33 mirr;
mirr << -1 << 0 << 0 << endr
     <<  0 << 0 << 1 << endr
     <<  0 << 1 << 0 << endr;
vec3 omegaM;

for(i = 0; i < 3; i++){
        for(j=0;j<3;j++){
                hp.mc(i,j)= hp.hT(i,j);
        }
}

hp.mc = mirr*hp.mc;

dRm = inv(hp.mc)*hp.ml;

//      cout << hp.mc << "\n";
//      cout << dRm << "\n";
//      cout << hp.sc << "\n";
//      cout << dRs << "\n**************************\n";

CalculateOmega(dRm,&omegaM);
```

```
                hp.ml = hp.mc;

//              cout << "This is my marker ....\n";

                rt_mutex_acquire(&hp.transf_lock,TM_INFINITE);
                hp.transf = hp.ur_T;
                rt_mutex_release(&hp.transf_lock);

                rt_mutex_acquire(&hp.v_lock,TM_INFINITE);

                v_M = eye(4,4);

//              printf("F : [%6.4f, %6.4f, %6.4f] - v: [%6.4f, %6.4f, %6.4f] \n", \\
                        hp.f[0], hp.f[1], hp.f[2], hp.v[0], hp.v[1],hp.v[2]);

                for(i=0;i<3;i++){

                        v_M(i,3) = hp.ur_T(i,3) + hp.v[i]*SPEIL(i,i);
                }
                rt_mutex_release(&hp.v_lock);

                rt_mutex_acquire(&hp.T_lock,TM_INFINITE);

                Xh = my_Trans_Inv*inv(hp.hT)*hp.haptic_bias*my_Trans;
```

```
set_T = v_M*Xh*hp.robot_bias;
rt_mutex_release(&hp.T_lock);

for(i = 0; i < 3; i++){
        for(j=0;j<3;j++){
                hp.sc(i,j)= ur_T(i,j);
        }
}

dRs = inv(hp.sc)*hp.sl;
vec3 omegaS;
CalculateOmega(dRs,&omegaS);

hp.sl = hp.sc;



ur_control_position_haptic(set_T, q_meas, hapticgain, &qd);

getJacobian_0_T(data.status.q, &J_bb);
v_c = J_bb*qd;

f_sum = pow(f_filt[0] + f_filt[1] + f_filt[2],2);

vector2array(&qd, cmd.qd, 6);
```

```
                        logg(   F[0],F[1],F[2],f_filt[0],f_filt[1],f_filt[2],
                                hp.hdv[0], hp.hdv[1], hp.hdv[2], hp.v[0], hp.v[1], hp.v[2],
                                set_T[12], set_T[13], set_T[14], hp.my_f[0], hp.my_f[1], hp.my_f[2],
                                hp.p1[0], hp.p1[1], hp.p1[2], hp.p2[0], hp.p2[1], hp.p2[2],
                                omegaM(0), omegaM(1), omegaM(2), omegaS(0), omegaS(1), omegaS(2),
                                hp.EinS[0], hp.EinS[1], hp.EinS[2], hp.EoutS[0], hp.EoutS[1], hp.EoutS[2]
                                );

                        for(i=0;i<6;i++){
                                if(hp.bt){cmd.qd[i] = 0;}
                                else{hp.ur_T = set_T;}
                        }

                        myC++;

                        if ( data.cmd.controller == UR_CONTROLLER_EXTERNAL){

                                int write_return = rt_queue_write(&ctrl_queue, &cmd, sizeof(cmd), Q_NORMAL);
                        }
                }
        clock_gettime(CLOCK_REALTIME,&ts);
        current = ts.tv_nsec;
//      cout << "Robot time is: " << current << " - " << last << " = " \\
                << (current - last) << "Max: " << max << "\n";
```

```
        robot_time_logg(((current-last)/CLOCKS_PER_SEC)*1000*1000);

        if(max < (current-last)){max = current - last;}
//      printf("Robot loop: %0.4f ms\n", (((double)clock()-start) / CLOCKS_PER_SEC)*1000*1000);
        }

}

void ur_external_controller_cleanup(int i){

        running = 0;
        rt_task_join(&rt_external_robot_controller);
        rt_task_join(&rt_external_haptic_controller);

}

int main(int argc, char* argv[])
{

        mlockall(MCL_CURRENT | MCL_FUTURE);

        signal(SIGTERM, ur_external_controller_cleanup);
        signal(SIGINT, ur_external_controller_cleanup);

        running = 1;
```

```
hapticgain << 3.5 << 3.5 << 3.5 << 3.5 << 3.5 << 3.5 << endr;
hp.haptic_bias = eye(4,4);
hp.robot_bias = eye(4,4);
hp.hT = eye(4,4);
hp.bt = 1;
hp.mc.eye();
hp.ml.eye();
hp.sc.eye();
hp.sl.eye();
hp.wall = {0,0,0};

initialize_logg();
initialize_time_logg();

hp.vsum.zeros();

rt_mutex_create(&hp_lock,"HP_LOCK");
rt_mutex_create(&hp.v_lock,"v_lock");
rt_mutex_create(&hp.hdv_lock,"hdv_lock");
rt_mutex_create(&hp.hdlv_lock,"hdlv_lock");
rt_mutex_create(&hp.f_lock,"f_lock");
rt_mutex_create(&hp.hdf_lock,"hdf_lock");
rt_mutex_create(&hp.T_lock,"T_lock");
rt_mutex_create(&hp.transf_lock,"transf_lock");

int rt_error = ur_linux_cpuaffinity_all(8);
```

```
printf("CPU affinity returned %i\n", rt_error);

if (InitializeHapticDevice() < 0){return -1;}

int error2 = rt_task_spawn(&rt_external_haptic_controller, "haptic_ctrl",0,70, \\
                T_JOINABLE,&rt_task_external_haptic_controller,NULL);

if(error2 < 0){printf("%s(%i): Haptic rt_task_spawn had error: %s\n",__FUNCTION__, \\
                __LINE__,strerror(-error2)); return 0;}

else {printf("Haptic RT task spawned.\n");}

int error1 = rt_task_spawn(&rt_external_robot_controller, "robot_ctrl",4*1024*1024,70, \\
                T_JOINABLE,&rt_task_external_robot_controller,NULL);

if(error1 < 0){printf("%s(%i): Robot rt_task_spawn had error: %s\n",__FUNCTION__,__LINE__, \\
                strerror(-error1)); return 0;}

else {printf("Robot RT task spawned.\n");}

rt_task_join(&rt_external_haptic_controller);
rt_task_join(&rt_external_robot_controller);

if (StopHapticDevice() < 0){return -1;}

return 0;
```